Trabajo Fin de Máster
"Máster Universitario en Microelectrónica: Diseño y Aplicaciones de Sistemas Micro/Nanométricos"

Análisis de técnicas de routing diferencial en CriptoASICs: Adecuación del proceso previo de Place & Route

Adrián Guijarro Córdoba

Julio 2020

TRABAJO FIN DE MÁSTER

Autor:

Adrián Guijarro Córdoba

Tutores:

D. Antonio José Acosta Jiménez

Dña. Erica Tena Sánchez

ÍNDICE

1 INTRODUCCIÓN

Basta con mirar a nuestro alrededor o incluso en nuestros bolsillos para ser conscientes que hoy en día los dispositivos electrónicos son parte de nuestra vida y los tenemos tan presentes, que los utilizamos prácticamente durante las 24 horas del día. Desde dispositivos que utilizamos para nuestro ocio, como los que nos ayudan a unificar una gran cantidad de herramientas en una sola, pasando por los que se utilizan para mover grandes cantidades de dinero. Tanto una Smart tv, un smartphone o una tarjeta de crédito, manejan una gran cantidad de información que, en muchos casos, queremos que sea segura y secreta frente a terceros, permitiendo así mantener nuestra privacidad a salvo.

Esta seguridad de los dispositivos electrónicos está basada en muchos casos en criptografía, algo que cada vez es más recurrente debido a la cantidad de información que se maneja electrónicamente y la necesidad de la sociedad de mantener esta información incólume.

Los circuitos criptográficos se componen tanto de un hardware como de un software específico, que se complementa entre sí y que está diseñado con mucho cuidado por una amplia y experimentada comunidad de científicos, ingenieros y desarrolladores. Sin embargo, no basta con tener un algoritmo extremadamente seguro o unos componentes cuya principal característica sea la robustez, sino que hay que dedicarles también especial interés y cuidado a los llamados ataques laterales o *Side Channel Attacks* (SCAs), donde, en algunas de las contramedidas, las condiciones de simetría son muy importantes y el interconexionado adecuado de los componentes tiene un lugar destacado para no comprometer la seguridad del criptosistema.

2 MOTIVACIÓN Y OBJETIVOS

Dentro del mundo de la criptografía, uno de los desafíos más complejos para el diseño de hardware seguro son los anteriormente citados ataques laterales, en adelante SCA. Los SCA se basan en medidas físicas como el consumo de potencia o retraso de señal entre entrada/salida realizadas sobre el cifrador para la obtención de la información y presentan la ventaja (o desventaja), de que son no invasivos y necesitan muy poco material, consiguiendo así presentar una amenaza real para circuitos en los que los chips de seguridad son fácilmente observables, como pueden ser por ejemplo las tarjetas de crédito.

De entre los SCAs existentes, el que podría definirse como particularmente efectivo es el análisis de potencia diferencial o *Differential Power Analysis* (DPA). Estos ataques están basados en la relación de dependencia existente entre los datos procesados y el consumo de potencia en CMOS. Con ellos, y gracias a un análisis estadístico, pueden descubrir información secreta. Esta relación de dependencia está basada en que, pensando en cualquier puerta del circuito, pueden ocurrir 4 transiciones en la señal de salida:

- Cambio de valor bajo (0) a valor alto (1)
- Cambio de valor alto (1) a valor bajo (0)
- Mantenerse en valor bajo (0)
- Mantenerse en valor alto (1)

De las situaciones mencionadas, en las dos primeras se producen consumos dinámicos de potencia. De este modo, si conocemos el consumo de potencia en un punto y hacemos un análisis estadístico, podemos descifrar el contenido secreto [1].

No obstante, existen medios para hacer frente a estos ataques. Hay diversas técnicas para hacer un diseño más robusto, destacando las centradas en ocultar y enmascarar el consumo de potencia asociado al dato procesado. Las técnicas de enmascaramiento basan su operación en enmascarar la relación existente entre el procesado de datos y el consumo utilizando una señal de enmascaramiento. Esto permite que el consumo de potencia sea independiente de los valores, incluso para dispositivos con consumo de potencia dependiente de datos.

Por otro lado, las técnicas de ocultamiento, consisten en romper la relación entre el consumo de potencia y los datos procesados. Para ello se aplican dos técnicas diferentes, bien hacer ver al atacante un consumo de potencia constante independientemente de las transiciones existentes, o bien hacer un consumo de potencia aleatorio independiente del procesado de datos, obteniendo por tanto inmunidad frente a este tipo de ataques [2].

Algunas técnicas de ocultamiento se sirven de las lógicas de doble rail (DPL: Dual-Rail with Precharge Logic) para conseguir un consumo de potencia independiente del dato procesado. No obstante, además de la lógica adecuada, es necesaria una máxima simetría que se consigue aplicando técnicas de routing diferencial para disminuir aún más la vulnerabilidad frente a los ataques, y es aquí donde se encuentra el principal objetivo de este trabajo final de máster, que consiste en analizar y evaluar los procedimientos existentes para routing diferencial en ASICs utilizando lógica DPL para su posterior aplicación a un ejemplo criptográfico.

Para ello se ha realizado una investigación sobre las lógicas de doble rail existentes y aplicables a ASICs y se han detallado sus características, especificando ventajas e inconvenientes, como resulta en el capítulo 3. Se selecciona la técnica WDDL por su facilidad para ser implementada con celdas estándar en un proceso de síntesis digital convencional.

En el capítulo 4 se concentra el análisis y evaluación de las diferentes técnicas y procedimientos existentes para realizar routing diferencial, finalizando con una comparación entre dichas técnicas y una propuesta de técnica para utilizar en el ejemplo.

Posteriormente, en el capítulo 5, se ha realizado el diseño lógico de un ejemplo criptográfico (S-box 4 Piccolo), sobre el que aplicaremos la adecuación a la técnica de routing diferencial escogida.

A continuación, en el capítulo 6, se ha realizado una comparativa de diversos procesos de Place & Route para ver como afectaban al layout final la modificación de diversos parámetros, buscando obtener la máxima simetría en las señales de salida.

Con el fin de describir las metodologías y herramientas empleadas en este trabajo, se ha realizado el capítulo 7.

El capítulo 8 abarca las conclusiones extraídas del trabajo realizado, además de incluir una valoración personal.

En el capítulo 9 se incluyen las referencias manejadas para la obtención de información, así como imágenes y demás contenido utilizado en la elaboración de este trabajo.

Se han incluido una serie de agradecimientos, que se encuentran en el capítulo 10.

Finalmente, el Anexo 1 hace referencia al código utilizado para la implementación de la S-box 4 Piccolo en lógica WDDL, el Anexo 2 incluye el código utilizado como testbench para probar el circuito diseñado, el Anexo 3 contiene el código para el posicionado manual de los pines de entrada y salida del diseño y el Anexo 4 contiene la información sobre capacitancia, resistencia, producto RC, variación relativa y variación relativa media de las salidas de las celdas.

3 ANÁLISIS Y EVALUACIÓN DE LÓGICAS DPL APLICABLES A ASICS PARA HACER FRENTE A DPAS

El principal objetivo de las lógicas DPL es hacer frente a los ataques DPA, necesitando para ello una igualdad en el consumo de energía en las líneas de estos circuitos, haciendo que el consumo de potencia no dependa de las transiciones de los valores. Las lógicas DPL operan en dos fases: precarga y evaluación, y se aplica en señales de doble rail, es decir, en cada celda se proporciona una salida determinada X y su complementaria $\bar{X}$. La fase de precarga es utilizada como inicialización del circuito, ya que establece unos valores determinados a las puertas lógicas y con ello permite comenzar nuevos procesos desde un estado eléctrico conocido, evitando así transiciones inesperadas entre dos estados. Esta fase de precarga puede hacerse tanto a 0 como 1, fijando para (X, $\bar{X}$) los valores a (0,0) o (1,1), dependiendo de cómo se realice el diseño. Por otro lado, la fase de evaluación consiste en el proceso en el que se evalúan las transiciones de los valores. Como previamente el circuito se ha precargado por ejemplo a valores de (0,0), si se ha realizado correctamente el diseño, con el consumo de potencia no se puede saber si el dato de salida ha pasado a ser (1,0) o (0,1), evitando poder detectar cuál de las transiciones se ha realizado y ganando robustez frente a DPA.

A pesar de la idealidad de la teoría, esto tiene que ser implementado y, como cabe esperar, no es perfecto. Esta lógica también presenta vulnerabilidades que, si las basamos en el ámbito temporal, pueden ser glitches, como por ejemplo transiciones falsas, eliminando así el consumo constante, o efectos de pronta propagación (fenómeno que consiste en que, si las entradas llegan a una puerta OR a diferente tiempo, en cuanto una de las entradas es 1, independientemente del valor de la otra, el valor de la salida es 1) [3].

Estas no idealidades también están presentes en el ámbito capacitivo. Para que el consumo de potencia sea constante, el valor de capacitancia total también debe serlo. La capacitancia total está compuesta de cuatro componentes: Capacitancia interna de los nodos, capacitancia intrínseca de salida, capacitancia de interconexión y la capacitancia intrínseca de la carga. En el caso de ASICs que implementan su lógica con celdas estándar, a pesar de que estas no tienen la misma capacitancia interna e intrínseca y, suponiendo que las señales diferenciales tienen el mismo recorrido, puede asumirse que las capacitancias de interconexión, que son las dominantes, son equivalentes. Esto se debe a que, con la reducción de la longitud de canal de los transistores, las capacidades internas e intrínsecas pierden importancia respecto a las de interconexión [4].

Dentro de la lógica DPL, existen varios tipos o conceptos experimentados entre los que cabe destacar como conceptos de ocultamiento SABL [5] y WDDL [4], o bien como conceptos de enmascaramiento DRSL [6], MDPL [7] e iMDPL [8], todos ellos aplicables a ASICs. No obstante, existen otras técnicas o variaciones más utilizadas en otro tipo de circuitos como podrían ser DPL-noEE, AWDDL o DWDDL, las cuales son utilizadas sobre todo en FPGAs [9] o SecLib (Secure Library) [3].

De las lógicas anteriormente mencionadas y sus variantes, se ha realizado una investigación y análisis de las desarrolladas para ASICs, que son en las que se centra el trabajo, y se muestran a continuación.

3.1 SABL

SABL (Sense Amplifier Based Logic, [5]): Es un tipo de lógica que consigue cargar en cada ciclo todas las capacitancias con un valor constante para fijar la cantidad de energía que se descarga en cada transición. Esto lo consigue modificando el valor de la salida independientemente del valor o de la secuencia de entrada y manteniendo constante una capacitancia de carga, que es igual a la que tienen todos los nodos internos combinada con una de las cargas de salida balanceadas. Con una lógica diferencial oculta el valor de entrada, haciendo que el consumo de energía sea el mismo independientemente del valor. La puerta SABL está basada en el flip-flop StrongArm110 (SAFF), al que le han eliminado las puertas estáticas NAND del latch SR y le han incorporado una función lógica intercambiando el par diferencial de entrada por una red diferencial pull down (DPDN), que de forma genérica puede verse en la Ilustración 1. Entre las principales ventajas se podría destacar que tiene un consumo de corriente prácticamente constante. Por ejemplo, comparando esta lógica con la SC-CMOS, se reduce en hasta 40 veces las variaciones del consumo de energía. Sin embargo, este tipo de lógica lleva asociadas unas desventajas, ya que conlleva un aumento de área y consumo al menos del doble que la lógica SC-CMOS, además de necesitar un diseño full-custom [10], [11].

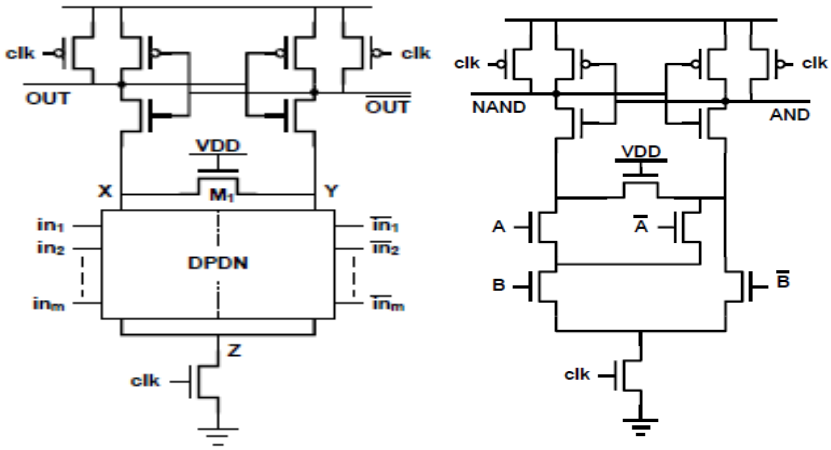


Ilustración 1. Celda SABL tipo n genérica (Izquierda) y puerta AND-NAND en SABL (Derecha) [10].

3.2 WDDL

WDDL (Wave Dynamic Differential Logic, [4]): Es una de las lógicas más comunes en DPL y es aplicable tanto para ASICs como para FPGAs. En ella únicamente se utilizan puertas positivas como AND/NAND y OR/NOR para asegurar la ausencia de glitches (para implementar NAND/NOR basta con intercambiar las salidas de una puerta AND/OR con su complementaria). En el caso de querer construir puertas XOR/XNOR, esto se realizará mediante dos puertas AND/NAND y una OR/NOR. Utiliza pares de puertas dobles para ocultar y asegurar actividad constante en los que únicamente se precarga el primer valor con (0,0) y se propaga al resto de puertas, siendo de este modo la capacitancia dominante la de interconexión. Dentro de la lógica WDDL, existen variantes como pueden ser AWDDL (Asynchronous Wave Dynamic Differential Logic), DWDDL (Double Wave Dynamic Differential Logic), WDDL con duplicación de backend dividida, IWWDL o WDDL w/o EE (WDDL sin efecto de pronta propagación) [4]. Esta lógica presenta ciertas ventajas, ya que además de tener unas características similares a SABL, es una lógica sencilla que utiliza celdas estándar, evitando el diseño full-custom. Por el contrario, como punto negativo se puede mencionar que sufre el llamado efecto de pronta propagación. Esto causa consumo de energía dependiente de datos y por lo tanto fugas de corrientes de canal, por lo que se necesitan señales perfectamente

balanceadas [3]. Aunque más adelante se explicará la implementación de esta lógica con mayor detenimiento, en la Ilustración 2 podemos observar un ejemplo de un circuito WDDL, en el que se ha destacado una puerta AND con este tipo de lógica.

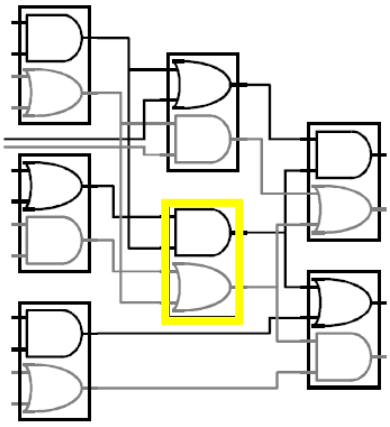


Ilustración 2. Circuito WDDL con puerta AND destacada [4].

3.3 SecLib

SecLib (Secure Library, [3]): Este tipo de lógica está basada en primitivas asíncronas cuasi insensibles a retrasos. Está balanceada de modo que proporciona valores constantes de tiempo y disipación de potencia en las fases de precarga y evaluación, proporcionando así una mayor resistencia frente ataques que si se compara con las características presentes en WDDL. Sin embargo, requiere más área y es una lógica full-custom [12]. Un ejemplo de esta lógica lo encontramos en la Ilustración 3, donde se muestra un esquemático de una puerta AND en esta lógica junto con su arquitectura interna.

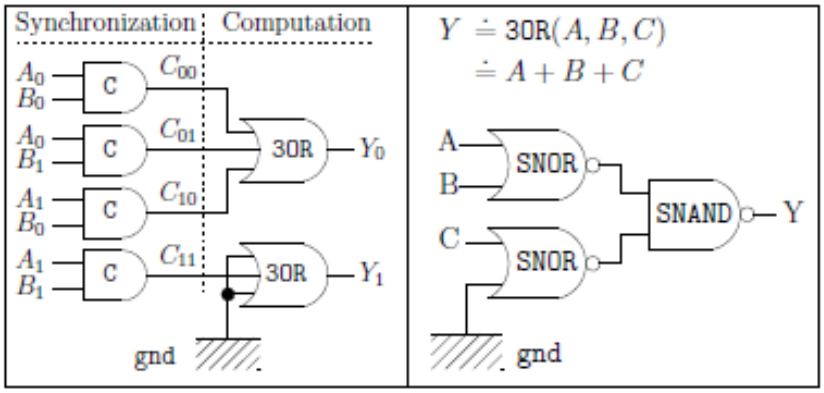


Ilustración 3. Esquemático de puerta AND en SecLib (Izquierda) y su arquitectura interna de las 3OR (Derecha) [3].

3.4 MDPL

MDPL (Masked Dual-Rail Pre-charge Logic, [7]): Lógica dual que utiliza el enmascaramiento para evitar los posibles glitches. A diferencia de otras lógicas en los que se busca balancear perfectamente la potencia consumida en todas sus conexiones, esta lógica aleatoriza las señales de enmascaramiento en el circuito. Del mismo modo consta de dos fases: Precarga a 0 y evaluación, y utiliza puertas AND/NAND y OR/NOR para implementar sus funciones, basando su circuito en la función majority (MAJ), como se muestra en la Ilustración 4. Entre sus principales ventajas se encuentra que puede implementarse con celdas CMOS estándar y no tiene restricciones de Place & Route, ya que, a pesar de necesitar una señal complementaria, estas no deben estar balanceadas,

gracias a la inclusión de la señal de enmascaramiento. Sin embargo, presenta algunos inconvenientes. Esta lógica tiene una disipación de energía predictible cuando las señales de puerta no están sincronizadas y presenta fugas de corriente debido a su asincronismo. Además, necesita una mayor cantidad de energía y área y la velocidad del circuito es relativamente lenta comparado con otras lógicas.

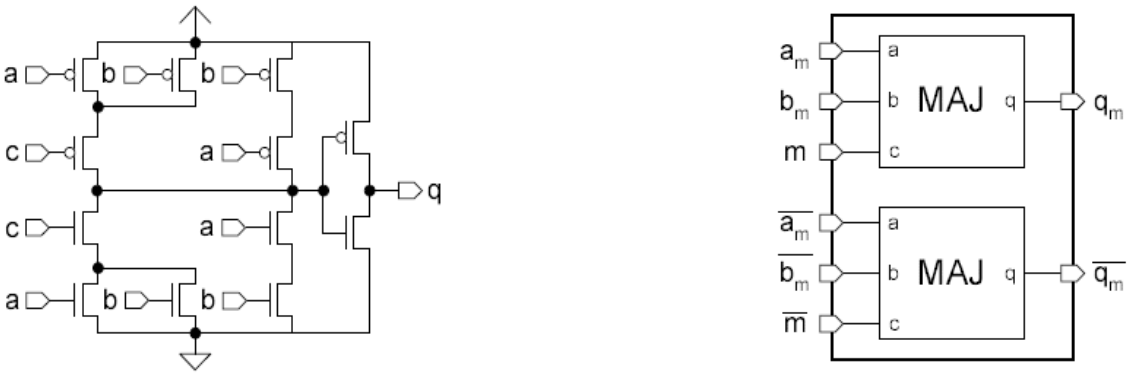


Ilustración 4. Esquemático de puerta majority CMOS (Izquierda) y de una puerta AND en MDPL (Derecha) [7].

3.5 iMDPL

iMDPL (Improvements of MDPL, [8]): Es una lógica similar a MDPL, ya que la utiliza como base, pero añadiendo mejoras que evitan el efecto de pronta propagación basadas en una unidad de detección de evaluación-precarga (EPDU), como la resincronización de todas las entradas incluyendo Latches-SR para controlar el comienzo de las fases de precarga y evaluación. Un ejemplo de su implementación se puede observar en la Ilustración 5, donde se muestra la estructura de una puerta AND. Su principal ventaja frente a su predecesor MDPL, es que mejora el efecto de pronta propagación y las corrientes de fuga, aunque presenta un comportamiento dependiente de datos y fugas de corrientes debido a que sus señales de enmascaramiento no están balanceadas. Si lo comparamos con MDPL, necesita 3 veces más área, es 3 veces más lento y consume un 50% más de energía, por lo que su uso está recomendado únicamente para circuitos críticos [13].

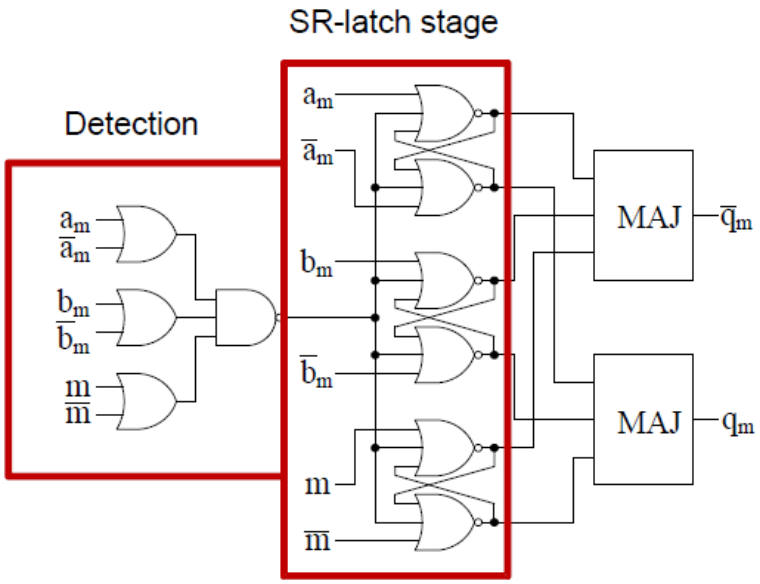


Ilustración 5. Estructura de celda AND en iMDPL [13].

3.6 DRSL

DRSL (Dual-Rail Random Switching Logic, [6]): Es una lógica basada y derivada de RSL (Random swithching logic) y MDPL, con la salvedad de que esta, emplea señales sincronizadas. Para ello utiliza un circuito independiente en cada puerta que crea una fase de precarga para sincronizar las señales (las originales y las de enmascaramiento, esta última para eliminar restricciones de routing) y otra de evaluación, siendo ambas independiente de los datos. Consta de 6 señales como mínimo, que son las dos entradas comunes, una entrada de enmascaramiento (utilizada para eliminar restricciones en el routing), y sus tres complementarias, como se observa en el ejemplo de la Ilustración 6, donde se muestra tanto una puerta NAND como AND. Tiene la ventaja de que puede utilizar puertas XOR e incluso inversores y consigue reducir la mayoría de fugas en el canal, siendo más robusta que otras lógicas. Por ejemplo, reduce las fugas en un 67% respecto a lógica WDDL y un 78% respecto a la lógica MPDL. Por otro lado, las puertas de esta lógica no son estándar, y necesitan una cantidad de área mucho mayor, implicando un incremento de área en sus puertas lógicas de entre 2 y 7 veces respecto a puertas lógicas estándar [6].

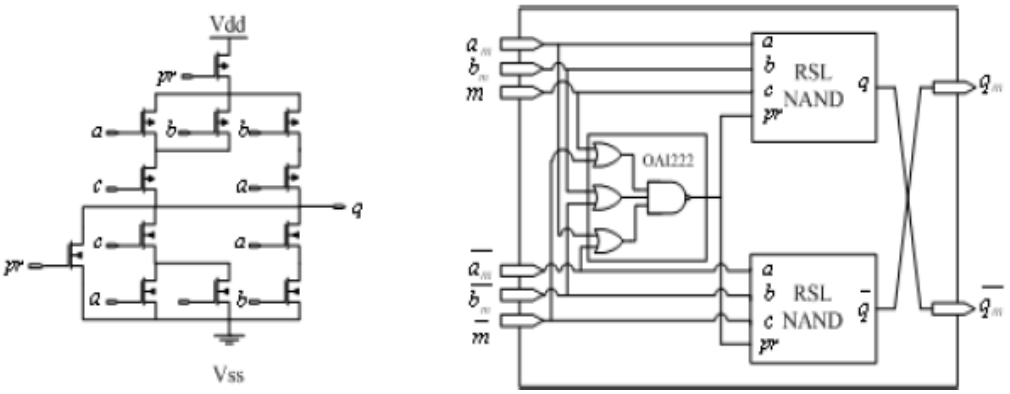


Ilustración 6. Puerta NAND en RSL (Izquierda) y puerta AND en DRSL [6].

3.7 STTL

STTL (Secure Triple Track Logic, [14]): Se trata de una lógica que se sirve de esperar a que todas las señales sean válidas o nulas para hacer la precarga o la evaluación, evitando de este modo cualquier riesgo de sufrir glitches, lo que conlleva una ralentización del proceso debido a la espera. Para ello incluye una señal de sincronización o validación, cuyo routing debe realizarse de modo que sea más lenta que las señales de doble rail. Otro de los posibles inconvenientes, es que se trata de una netlist heterogénea, ya que incluye tanto señales diferenciales como simples, por lo que como veremos en el siguiente capítulo, se hace más difícil el layout [3]. Un ejemplo de esta lógica se muestra en la Ilustración 7, que contiene el diseño de una puerta AND en STTL.

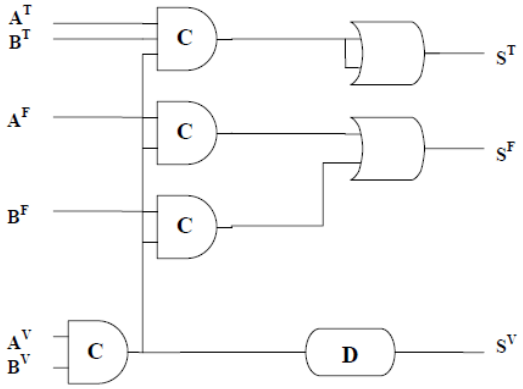


Ilustración 7. Puerta AND en STTL [14].

A modo resumen, se ha creado la Tabla 1, en la que se especifican las principales características de cada una de las lógicas que se han explicado anteriormente.

En esta tabla, se incluye el tipo de lógica de la que se trata en primer lugar. En la segunda columna se hace referencia a una señal de enmascaramiento presente y que, obviamente, solo se encuentra en las lógicas de enmascaramiento. La existencia de una señal de enmascaramiento permite eliminar el desequilibrio, pero resulta en un incremento de área considerable. A continuación, en la tercera columna, se detalla cuál de las lógicas tiene señal de sincronización tanto para la fase de precarga, como para la de evaluación, evitando así la existencia de glitches. En la siguiente columna se añaden cuáles son los requisitos de la lógica, dividiendo entre los requisitos principales, y los necesarios al realizar el proceso de back-end. En la última columna se añade el desequilibrio entre las líneas diferenciales, incluyendo la carga, y diferencias de interconexión y estructura. Esto es una fuente de información sobre pérdidas y cuanto más bajo sea, más seguro resulta.

Lógica	Señal de enmascaramiento	Sincronización		Requisitos		Desequilibrio
		Precarga	Evaluación	Principales	Back-end	
SABL	×	×	×	SAFF & DPDN *	Place & Route balanceado	BAJO
WDDL	×	×	×	Función positiva	Place & Route balanceado	ALTO
SecLib	×	✓	✓	Librería específica	Duplicación de Back-end	MUY BAJO
MDPL	✓	×	×	Puerta *Majority*	NO	NO
iMDPL	✓	×	×	Puerta *Majority*	NO	NO
DRSL	✓	✓	×	NO	NO	NO
STTL	×	✓	✓	NO	Retraso en señal de sincronismo	MUY BAJO

Tabla 1. Resumen características de lógicas.

* SAFF: Sense amplifier half of the StrongArm110 flip-flop.
* DPDN: Differential pull down network.

4 ANÁLISIS Y EVALUACIÓN DE TÉCNICAS Y PROCEDIMIENTOS EXISTENTES PARA REALIZAR ROUTING DIFERENCIAL

Tras haber hecho un análisis de las lógicas DPL más conocidas y utilizadas, existen diversos métodos de routing para conseguir un óptimo balanceo de las salidas en las lógicas de doble rail, ya que, un buen diseño DPL con un routing diferencial asimétrico es más vulnerable frente a ataques DPA. Dentro de las diversas técnicas de routing diferencial, habría varias a destacar.

4.1 Fat Wire

Esta técnica de routing diferencial consiste en posicionar cada par de líneas diferenciales como una única línea (con la característica de tener el espesor de 2 líneas paralelas determinadas) y, posteriormente dividir esa línea gruesa en dos líneas diferenciales, tal y como se muestra en la Ilustración 8. El punto central de la línea gruesa principal es el punto central existente entre las dos líneas diferenciales. El ancho de la pista principal (Wf) viene dado por la suma de dos veces la mitad del ancho de la pista diferencial (Wn) más el espaciado existente entre las líneas diferenciales (Pn): Wf= Pn+2*Wn/2. El espaciado entre pistas principales (Pf) está determinado por la suma de dos veces la mitad del ancho de la pista principal (Wf) más la distancia entre líneas principales (Δ): Pf=2*Wf/2+(Δ) [15], [16].

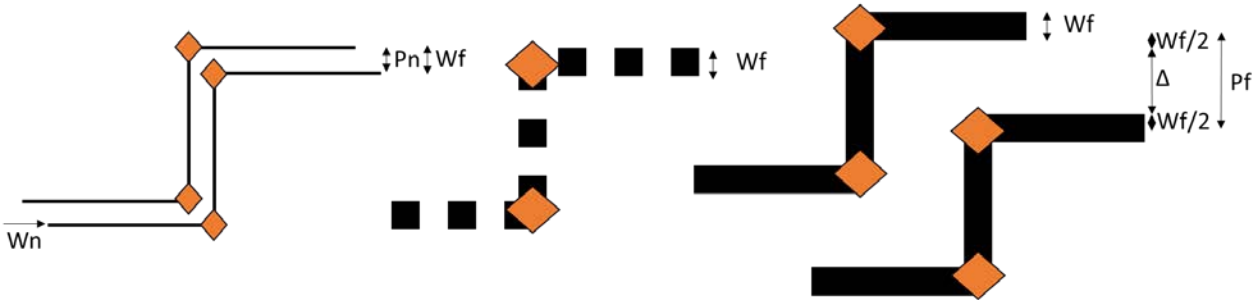


Ilustración 8. Medidas conversión Fat Wire.

4.1.1 Transformación

Para aplicar esta técnica lo que se realiza es el posicionamiento y routing con líneas principales, y posteriormente estas líneas se transforman a líneas diferenciales. Esta transformación consiste en transformar la línea principal en dos líneas diferenciales y la reducción de ancho de la línea principal al de las líneas diferenciales. Para ello se mantiene el centro de la línea principal como el centro de las dos líneas diferenciales, siendo generadas las líneas diferenciales una en el eje positivo y la otra en el negativo, siendo así tanto en el eje X u horizontal como en el eje Y o vertical. En la Ilustración 9 se representa esta transformación.

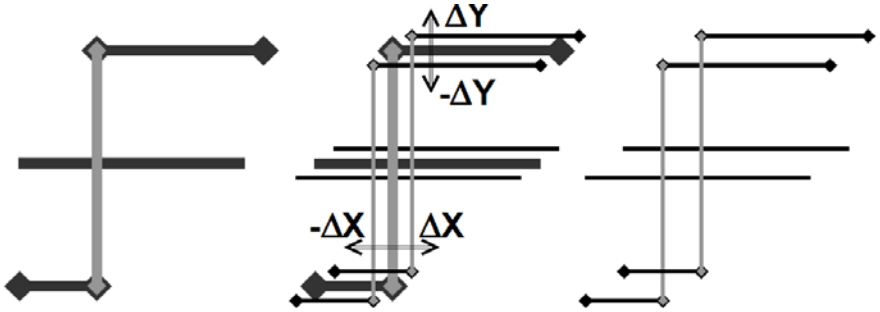


Ilustración 9. Línea principal (Izquierda); Operación de traducción (Centro); Líneas diferenciales (Derecha) [15].

Como se puede observar en la Ilustración 9, se genera un desplazamiento de ±ΔX y ±ΔY de las líneas diferenciales respecto de la línea principal. Además, es necesario mantener el mismo número de vías por cada segmento para controlar que la impedancia se mantenga igual en el par. Cada parte o segmento de la línea diferencial tiene la misma longitud y están posicionadas en la misma capa de metal, proporcionando así un equilibrio de las impedancias de las líneas diferenciales y de sus capacitancias parásitas respecto a otras líneas o el propio sustrato.

El proceso para realizar este tipo de diseño consiste en varios pasos.

1. Se diseña el circuito con un HDL, por ejemplo, Verilog y se generan unas especificaciones para líneas y vías simples que se almacenan en una librería. (fat.v & fat.lef respectivamente)
2. Se procede al posicionado de las puertas y el routing con líneas simples. (fat.def)
3. Se realiza la traducción a líneas diferenciales. (diff.def)
4. Junto con la librería de especificaciones para líneas y vías diferenciales (diff.lef) se finaliza el diseño. (diff.gds)

A continuación, se resume este proceso en la Ilustración 10.

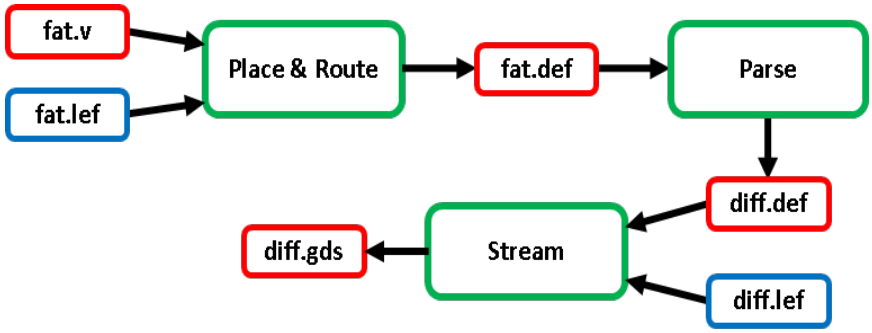


Ilustración 10. Proceso de routing diferencial mediante Fat Wire.

4.1.2 Restricciones

Como se puede observar en la Ilustración 9, las vías deben estar alineadas de manera inclinada, no perfectamente verticales u horizontales. De modo que para que haya un buen equilibro de longitudes, impedancias y capacitancias, los pines de entrada y/o salida de las celdas tienen que encontrarse con la misma inclinación, de lo contrario, esta técnica no sería efectiva.

4.1.3 Recomendaciones

Según, y gracias a la experiencia de algunos investigadores, con el paso del tiempo se han ido generando una serie de recomendaciones a seguir para facilitar el diseño y la transformación [16].

- En lo referente al grid se aconseja que la celda tenga una altura y ancho múltiplo del espaciado entre líneas. Del mismo modo, aconsejan que el grid sea al menos del mismo tamaño que el espaciado de los pines de la celda, para conseguir un correcto trazado de las líneas. También es recomendable que el espaciado mínimo entre líneas principales sea del doble que el de las líneas diferenciales, para que luego no existan problemas al realizar la transformación.
- Respecto a las capas de metal en donde se realiza el routing de las líneas, al hacer la división de línea principal a líneas diferenciales, hay que tener en cuenta qué ocurre cuando las líneas sufren un cambio de dirección. Como vemos en la Ilustración 9, al hacer una rotación de la línea de 90º, si se mantuvieran en la misma capa se produciría un corto entre ellas. Es por

este motivo que hay que hacer un cambio de capa de conducción. La principal recomendación es mantener en la medida de lo posible cada capa para una dirección de líneas. Por ejemplo, podría mantenerse una capa interna para líneas horizontales y otra para líneas verticales.

- Hasta ahora, se ha tratado todo como si únicamente existieran líneas diferenciales, sin embargo, también es necesario tener en cuenta una traducción de un diseño en el que nos encontremos tanto con líneas diferenciales como con líneas simples. En este caso hay varias formas de proceder:
  1. Realizar las líneas simples que serán definidas como diferenciales, realizar la traducción y posteriormente añadir las líneas simples.
  2. Realizar las líneas simples, y posteriormente realizar las que serán definidas como diferenciales y realizar la traducción.
  3. Realizar todo el diseño como líneas simples, diferenciando en la definición de estas cuales serán simples y cuales diferenciales. Posteriormente se realiza la traducción y, tras haber definido correctamente nuestras líneas, el diseño quedará tal cual necesitamos. Esta última técnica no suele ser recomendada para diseños con pocas líneas diferenciales o con unas fuertes restricciones en lo que a área respecta.

Como ejemplo se muestra la Ilustración 11, en la cual se ha realizado un diseño que consiste en 6 puertas diferenciales aplicando esta técnica.

En la parte izquierda se encuentra el diseño con líneas simples (antes de hacer la traducción) y, a la derecha, se encuentra el diseño con líneas diferenciales (al haber realizado la traducción).

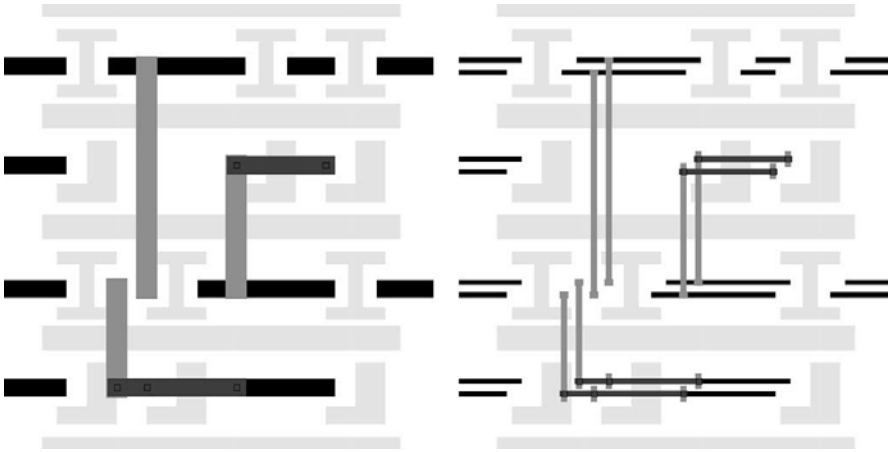


Ilustración 11. Ejemplo de diseño Fat Wire [16].

4.2 Backend duplication

Este método [17] consiste en realizar un flujo regular de back-end a partir de una netlist estándar (frente a una netlist diferencial), con la salvedad de que es necesario dejar espacio suficiente en el floorplan para realizar posteriormente una duplicación de la lógica del circuito. Para poder realizar esta duplicación simplemente se requiere que el resto de filas (capas de metal) no ocupadas, se mantengan libres, por lo que normalmente se obstruyen esas capas que queremos dejar para la duplicación.

El hecho de realizar una duplicación del circuito, no consiste únicamente en duplicar el circuito original, sino que es necesario intercambiar las celdas originales por sus complementarias, como podría ser el caso de una celda NAND-NOR, mostrado en la Ilustración 12.

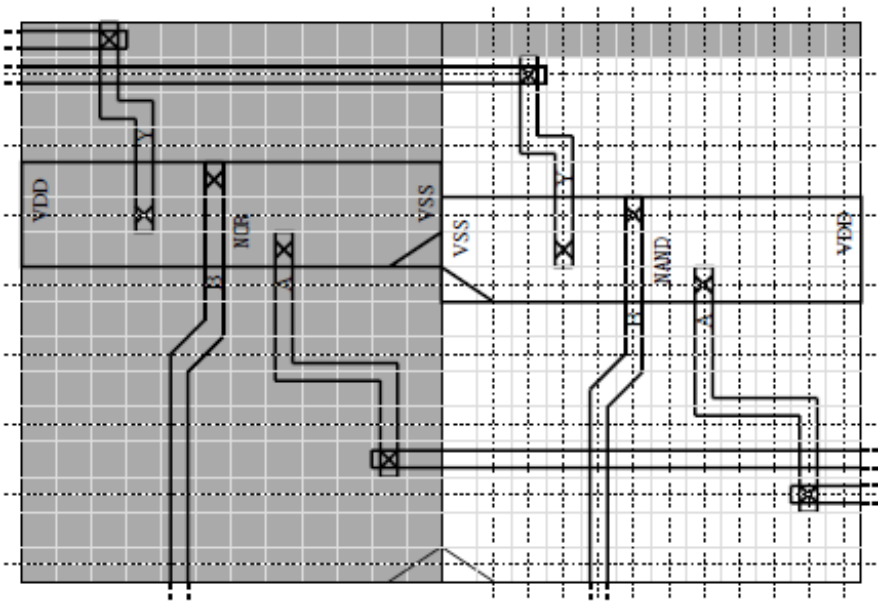


Ilustración 12. Diseño final de puerta NAND (y su dual NOR) mediante Backend Duplication. [17]

Posteriormente se procede a la interconexión. Para que no solo las celdas, sino las conexiones entre ellas puedan duplicarse, las líneas verticales que interconectan unas filas con otras, son forzadas a ocupar únicamente un canal de routing, obligando a las líneas verticales a ser perfectamente rectas. Esto asegura que el desplazamiento de una línea vertical una distancia equivalente al valor del pitch (separación entre pistas) no estaría causando un cortocircuito. En el caso de que las líneas verticales no fueran perfectamente rectas, se cruzarían con los espacios adyacentes que se han dejado libres para la duplicación de líneas verticales.

Esto no ocurre con las líneas horizontales, que permiten que no sean perfectamente rectas siempre y cuando se mantengan siempre dentro de la misma fila, ya que, al realizar la duplicación, la fila superior o inferior no se vería afectada.

4.2.1 Transformación

La transformación se lleva a cabo de modo que la traslación se realiza horizontalmente en orden de separación entre pistas (pitch) por razones de routing, y verticalmente en orden de alturas de filas por razones de posicionamiento. En resumen, la duplicación se realiza en los ejes (X, Y) con un vector de traslación tal que (espaciado de pistas, altura de fila).

Una vez se realiza esta transformación, se obtienen dos netlist idénticas entrelazadas entre ellas, las cuales no se puedes desvincular, ya que no son independientes, si no que algunas señales deben intercambiarse entre puertas contiguas, como se muestra en la Ilustración 13.

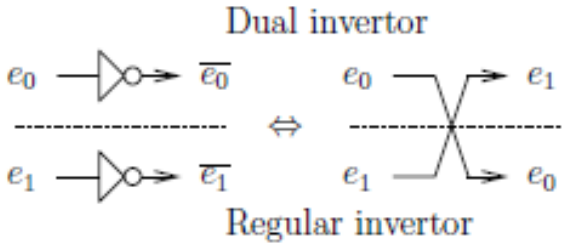


Ilustración 13. Inversión de señales sin inversor. [17]

4.2.2 Restricciones

Una de las principales restricciones que se deben cumplir es que finalmente, debe mantenerse la indistinguibilidad de las dos netlist. Por ejemplo, cuando se generan los dummies (trozos metálicos para cumplir con la densidad mínima de metal), estos deben generarse únicamente en las filas en las que está permitido el posicionamiento, por lo que hay que especificar restricciones y requisitos, evitando así que esto se realice en filas donde no se deberían colocar estos dummies. Tras esto, estas partes metálicas serán duplicadas y trasladadas según los requisitos que se han explicado en el apartado de *transformación*, de modo que no se generarán cortocircuitos.

Otras restricciones se han ido detallando intrínsecamente en el apartado general y en el de explicación. Estas restricciones referentes al posicionado, routing y duplicación, pueden automatizarse con un script que consiste en el **posicionamiento por bloques**, colocando una fila sobre otra lo más a la derecha posible y el **routing por bloques** de un canal sobre otro para canales verticales o bien para filas específicamente designadas como obstruidas, es decir, las que no permiten el posicionamiento de celdas, pero sí de metal.

En la Ilustración 14 se muestran estas restricciones en un plano de 2 partes de 16 filas y 12 columnas. En ella se puede observar cómo frente al método *Fat Wire*, estas restricciones son más flexibles, puesto que aquí, las líneas horizontales, no tienen por qué estar completamente rectas (pueden hacer zig-zag siempre que se mantengan en la misma fila), si no que esto solo se impone a las líneas verticales. Esto es una libertad considerable a la hora de realizar el diseño, permitiendo así un routing más rápido y confiable.

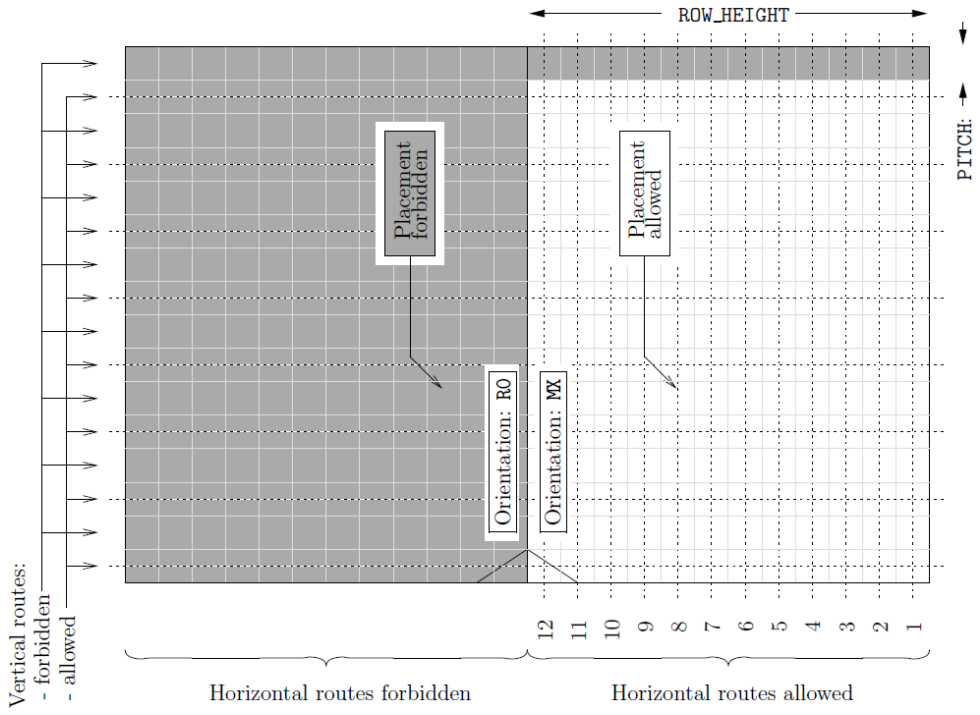


Ilustración 14. Plano de diseño de 16x12 [17].

Si comparamos este método con el regular, queda claro que no es necesario asignar o modificar guías de diseño, ya que se siguen las regulares, si no que únicamente es necesario añadir algunos pasos adicionales. Para representar gráficamente la diferencia entre el método *Backend Duplication* frente el método estándar, se ha generado la Ilustración 15.

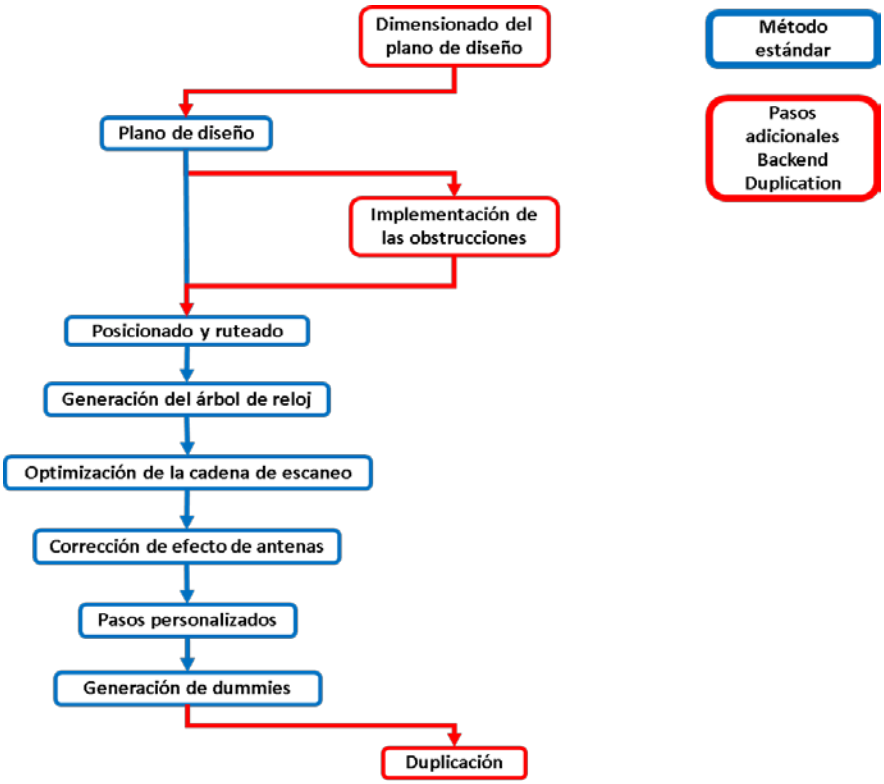


Ilustración 15. Backend Duplication vs Método estándar

Podemos observar cómo aplicar este método únicamente conlleva el añadir tres pasos, los cuales consisten en:

1. **Dimensionado del plano de diseño**: El plano de diseño está compuesto de dos partes. El núcleo, donde se colocan las celdas, y la oblea, donde además del núcleo se encuentra un canal extra que lo rodea, utilizado por ejemplo para el routing de un anillo de alimentación. Las dimensiones verticales y horizontales deben ser un múltiplo impar del número de filas y del espaciado entre líneas respectivamente para poder asegurar que, tras la duplicación, el posicionamiento y routing no excede el área del núcleo. En el caso de que se haya realizado el dimensionado incorrectamente, este puede repararse de modo que, si necesitamos una densidad de núcleo *d* y una ratio de aspecto *r*, antes de la duplicación deberíamos tener la densidad *d/2* y la ratio *r/2*. Si las dimensiones del núcleo previamente a la modificación eran (x,y), tras la modificación deberían quedar siendo:

$$x' = \left[\frac{x}{2 \cdot PITCH}\right] \cdot 2 \cdot PITCH, y' = \left[\frac{y}{2 \cdot ALTURA\_FILA}\right] \cdot 2 \cdot ALTURA\_FILA$$

2. **Implementación de las obstrucciones**: Como se ha descrito en el apartado de restricciones, es necesario especificar en qué filas se puede o no realizar el posicionamiento. Esto se hará en cuanto se haya generado el plano de diseño y justo antes del posicionamiento y routing.
3. **Duplicación**: El proceso de duplicación consiste en una traslación del posicionamiento seguida de un espejado horizontal de cada fila. Sin embargo, el routing es más complejo de duplicar, ya que un mero duplicado de las líneas de conexión, haría que algunas de estas líneas se salieran de la oblea, por lo que para evitar esto, los extremos de las líneas (u,v) deben ser tratados de modo que:
    a. Si (u,v) se encuentran en el núcleo -> (u', v') = (u + PITCH, v + Altura_filas).
    b. Cualquier otro caso (u', v') = (u, v).

Tras realizar estas tres transformaciones obtenemos como resultado la Ilustración 12.

4.2.3 Recomendaciones

En ocasiones es recomendable realizar la duplicación de líneas al realizar la netlist, de modo que se duplican todas las líneas y celdas. Con esto posibilitamos una verificación LVS (Layout Versus Schematic).

4.3 Divided backend duplication

Este método de layout diferencial está basado tanto en el método *backend duplication* como en la lógica DWDDL, la cual explicamos a continuación [18].

DWDDL es un tipo de lógica que consiste en una división de la lógica WDDL en dos partes. Una lógica WDDL en la cual no existe una inversión, consiste en dos partes que son duales, derivadas la una de la otra y en donde la diferencia reside en el intercambio de puertas AND por OR y viceversa, como se muestra en la siguiente Ilustración 16. No puede existir una inversión en lógica WDDL, ya que pararía la precarga; pero como es complicado realizar una lógica sin inversión, para solucionar esto basta con intercambiar una línea y su complementaria. Otra opción podría ser añadir una puerta XOR en vez del inversor, de modo que se invertirían las salidas manteniendo la señal de precarga. Para ello debe conectarse la señal de precarga a la puerta XOR que sustituye al inversor como se muestra en la Ilustración 17.

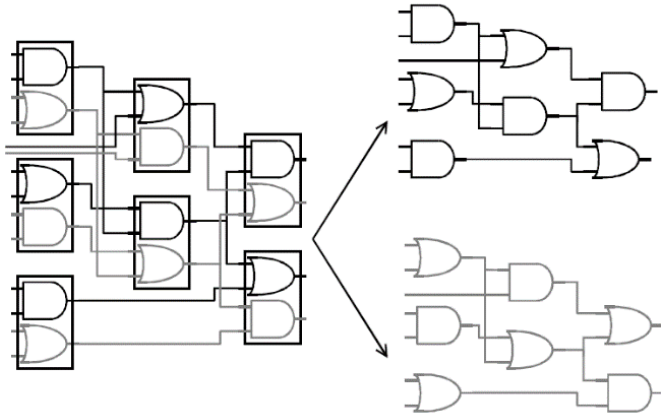
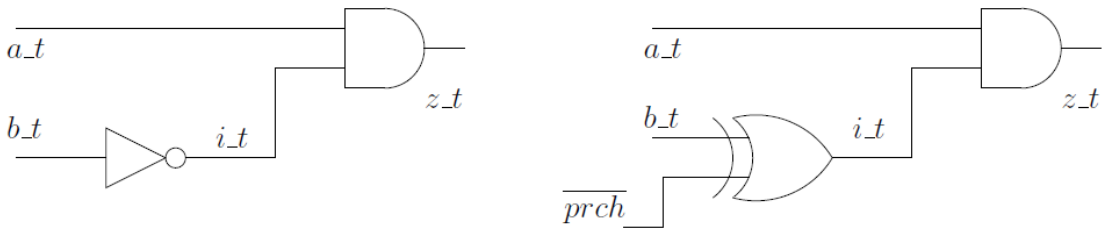


Ilustración 16. Derivación de DWDDL [4]



a_t	b_t	i_t	$prch$	z_t
0	0	1	1	0
1	1	0	1	0
0	0	1	0	0
0	1	0	0	0
1	0	1	0	1
1	1	0	0	0

a_t	b_t	i_t	$prch$	z_t
0	0	0	1	0
1	1	1	1	1
0	0	1	0	0
0	1	0	0	0
1	0	1	0	1
1	1	0	0	0

Ilustración 17. Sustitución de inversor por XOR [18].

Si en vez de unificar esta lógica dual WDDL la separamos, obtenemos la lógica DWDDL. Con esto, conseguimos evitar tener que balancear las señales diferenciales entre ellas, pero sin embargo es necesario balancear las interconexiones de las entradas a la lógica combinacional y generar registros compuestos, además de conllevar un incremento de área de 4 veces el original. Es una lógica que principalmente está diseñada para FPGA, sin embargo, también hay aplicaciones para ASICs [19].

4.3.1 Transformación

Tras tener nuestro diseño con puertas XOR en vez de inversores, el circuito puede implementarse separando la parte verdadera por un lado y por otro lado la parte complementaria, manteniendo comunes entradas y salidas. Una vista general del procedimiento sería la mostrada en la Ilustración 18.

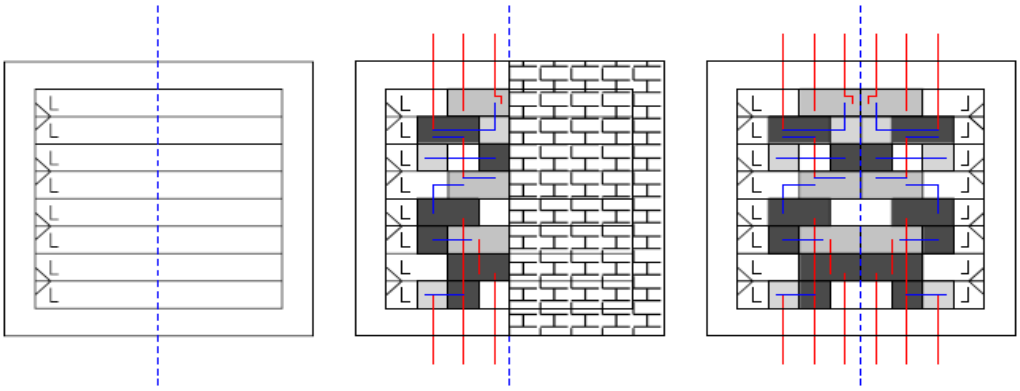


Ilustración 18. Vista general de Divided Backend Duplication [18].

En la Ilustración 18 se muestra en la izquierda el plano inicial. En la parte central, el plano una vez se ha posicionado y se ha realizado el routing de la parte verdadera del circuito dejando espacio para la parte complementaria. En la parte derecha, el circuito finalizado tras hacer la duplicación de la parte complementaria.

Este proceso de transformación puede describirse en 5 pasos y se muestra en la Ilustración 19:

1. Se procesa todo el circuito simple para reemplazar los inversores por puertas XOR.
2. Se realiza el floorplan de los componentes utilizando solo una mitad del plano de diseño, asegurando así espacio suficiente para la parte complementaria. Ilustración 18-izquierda.
3. La mitad del plano de diseño queda reservada a la parte complementaria. Ilustración 18-centro.
4. El diseño simple (sin señales diferenciales), se ha implementado correctamente.
5. Se implementa la parte complementaria haciendo una duplicación y traslación de los componentes del circuito.

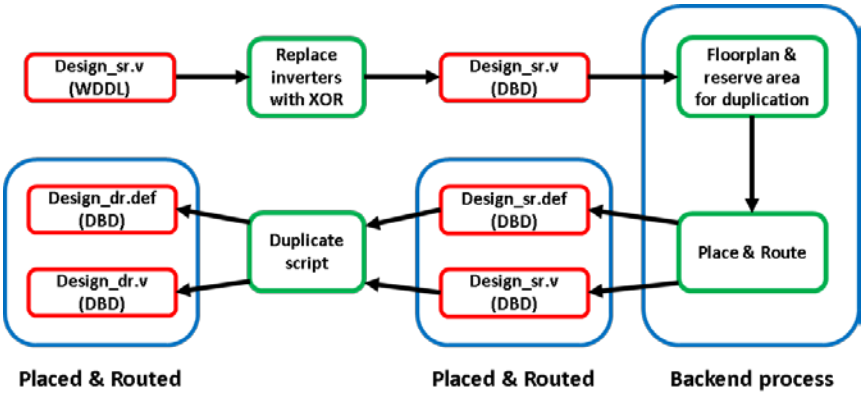


Ilustración 19. Visión general del método divided backend duplication.

Este método puede hacerse con pequeñas variaciones como pueden ser:

- En vez de hacer un *flip* a los componentes, desplazarlos.
- En vez de realizar el *flip* en el eje X, hacerlo en el Y.

4.3.2 Restricciones

Para la implementación se necesita que los pines de la parte complementaria sean los mismos, en la misma posición y capa de metal, y que el tamaño de las celdas complementarias sea el mismo.

4.4 Análisis crítico y selección de propuesta

Debido a las características de las diferentes técnicas de layout existentes, se ha realizado un análisis de las anteriormente mencionadas para compararlas entre ellas, y seleccionar la que se considera más adecuada para su implementación en un ejemplo explicativo.

4.4.1 Comparación entre alternativas

Si se realiza una comparativa entre *Fat Wire*, *Backend Duplication* y *Divided Backend Duplication* podemos encontrar que entre ellos existen diferencias considerablemente notorias.

Una diferencia se encuentra en que *Fat Wire* necesita hacer reglas de diseño específicas para el trazado de líneas, ya que, por ejemplo, para girar una línea necesita cambiar de capa y necesita redefinir el layout para acceder a los pines de una celda estándar, mientras que para los otros dos métodos no.

El *Fat Wire* necesita un diseño iterativo y repetitivo, de modo que al terminar el diseño se analizan las capacidades parásitas y se comparan, para que en el caso de que haya un mal balanceo de estas capacidades se modifique el layout hasta que las capacidades converjan en un balanceo correcto. Por el contrario, con el *Backend Duplication* y el *Divided Backend Duplication*, esto se realiza con el propio diseño, por lo que no es necesario iterar. Sin embargo, hay que tener en cuenta que, con estos dos, únicamente se pueden realizar diseños 100% diferenciales, mientras que el *Fat Wire* puede tratar con diseños que tengan tanto señales diferenciales como simples [17], [12].

Un ejemplo de implementación con diferentes metodologías se muestra en la Ilustración 20, donde se realiza la transformación de una puerta AND con *Fat Wire* (a) y con *Backend Duplication* (b).

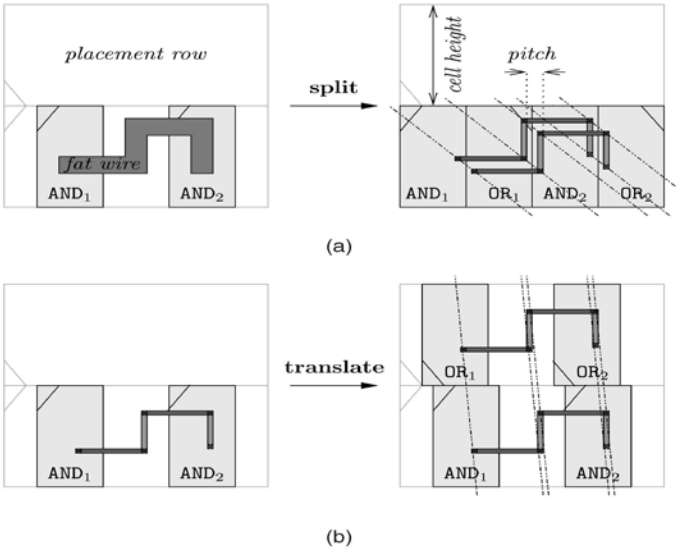


Ilustración 20. Fat Wire VS Backend duplication [12].

Otra diferencia de *Fat Wire* con *Backend Duplication* y *Divided Backend Duplication* es que estas dos últimas necesitan una traslación o reflexión especular respectivamente del diseño para trabajar con estos métodos, mientras que *Fat Wire* únicamente necesita tener cada celda al lado de la complementaria, pero no el diseño completo.

Entre las principales ventajas y diferencias que podemos encontrar entre el *Divided Backend Duplication* con los otros dos estilos, se encuentra el hecho de que, al poder separar la parte real de su complementaria, no tendremos problemas de acoplo capacitivo con otras líneas próximas al reducir el factor de escala. Con otros métodos, al reducir el factor de escala, como se mantienen las capacidades entre las líneas verdaderas y sus complementarias, la relación entre capacidades es mucho mayor, generando así una mayor asimetría.

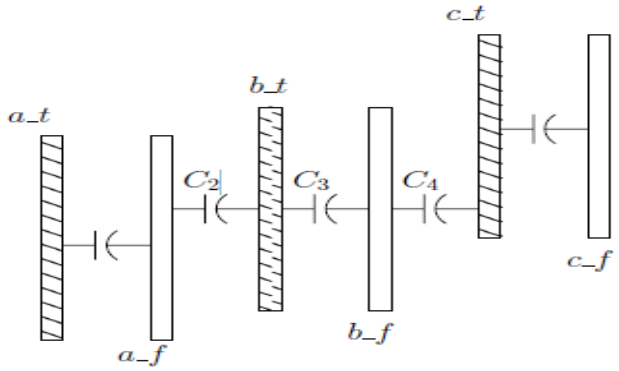


Ilustración 21. Capacidades de acoplo [18]

Podemos observar como en la Ilustración 21, existen unas capacidades entre las líneas a, b, c y sus complementarias o adyacentes. En el caso de necesitar reducir el ancho de pista debido a la tendencia a reducir el tamaño de los circuitos, la capacidad seguiría manteniéndose, por lo que la relación entre la capacitancia y el ancho se vería aumentada.

Otra ventaja asociada a este método respecto de las dos anteriores, y gracias a la separación de la línea simple y su complementaria, es que podemos procesar dos conjuntos de datos a la vez, y podemos actuar con un modo aleatorio, teniendo en una parte del circuito la señal real, y en la que se supone que debería ser la complementaria, señales aleatorias. Incluso se puede optar por eliminar la generación de señales de la parte complementaria cuando no queramos contrarrestar las medidas frente a un ataque porque no sean necesarias. Esto requiere únicamente cambiar las interfaces de entrada/salida al diseño dual [18].

Por otra parte, centrándonos en la inversión de la lógica, si hay señales diferenciales no es necesario utilizar un inversor, ya que en la lógica diferencial existe la señal positiva y la complementaria, por lo que basta con intercambiar señales para invertir el valor. Sin embargo, en el estilo *Divided Backend Duplication*, la lógica va separada y cabe la posibilidad de necesitar invertir una señal. Como se ha comentado, esto se hace intercambiando inversores por puertas XOR, pero al realizar esto, se genera un incremento de área y retraso de señal, que depende del circuito, en función de la cantidad de inversores que hayan sido necesarios reemplazar [18].

Para poder resumir estas diferencias de forma algo más visual, se ha realizado la Tabla 2.

Estilo	Reglas de diseño específicas	Necesidad de iterar	Mezcla de señales simples y diferenciales	Duplicación de diseño completo	Problema capacitivo al reducir factor de escala	Procesar diferentes conjuntos de datos a la vez*	Incremento de área y retraso de señal por intercambio de INV → XOR
Fat Wire	✔	✔	✔	✘	✔	✘	✘
Backend Duplication	✘	✘	✘	✔	✔	✘	✘
Divided Backend Duplication	✘	✘	✘	✔	✘	✔	✔

Tabla 2. Diferencias Fat Wire, Backend Duplication y Divided Backend Duplication.

* Procesamiento de conjunto de datos por separado, permitiendo actuar con modo aleatorio o eliminar parte complementaria de los datos.

4.4.2 Selección de propuesta

De las 3 técnicas anteriormente descritas, se ha seleccionado trabajar para la opción de la técnica *Fat Wire*. En concreto esta técnica ha tomado ventaja frente a las otras dos opciones, ya que tanto para el *Backend Duplication* como para el *Divided Backend Duplication* se requiere trabajar a nivel de celdas, mientras que con el *Fat Wire* se pueden tomar las celdas como unidad mínima, o se pueden tomar bloques más grandes, proporcionando mayor versatilidad a la hora de escoger la granularidad. Además, es importante tener en cuenta el hecho de que trabajar con *Backend Duplication* o *Divided Backend Duplication* obliga al posicionado manual de celdas, mientras que trabajar con *Fat Wire* puede proporcionarnos la facilidad de poder realizar el diseño automático o con una dedicación manual muy inferior en función de la unidad mínima de trabajo.

5 APLICACIÓN A UN EJEMPLO CRIPTOGRÁFICO SENCILLO

Como vehículo demostrativo, se ha elegido un circuito con una seguridad contrastada, de bajo consumo, buena aplicabilidad y con una complejidad media como es el cifrador Piccolo. Una de las partes más vulnerables frente ataques DPA son las S-Box (Substitution box), y esta será la que utilizaremos de forma aislada como caso de estudio. Este componente está involucrado en la relación existente entre los valores de texto plano y texto cifrado, de modo que debe ser lo más seguro posible.

Para analizar la aplicabilidad de la técnica de *Fat Wire* en este caso de estudio, se ha optado por la opción de adecuar el proceso previo de Place & Route tomando como unidad mínima la S-Box. Esta decisión viene marcada por una cuestión de granularidad que, de ser abordada en todos sus niveles, excedería con creces la dedicación de un Trabajo Fin de Máster:

- Adecuar la técnica de layout para trabajar con celdas estándar como unidad elemental para *Fat Wire* es posiblemente inviable por su complejidad y coste en área y recursos.
- Por el contrario, tomar como unidad mínima un bloque mayor que esta S-Box ha dado a pensar que los resultados pueden no ser concluyentes por problemas de extensión del circuito, ya que existiría una gran cantidad de layout asimétrico dentro de la unidad mínima y el trabajo realizado sería poco influyente.

5.1 S-Box 4 Piccolo

Piccolo es un cifrador de bloque de 64 bits tanto de entrada como de salida que es capaz de soportar claves de 80 y 128 bits, referidos como Piccolo-80 o Piccolo 128 respectivamente. Sin embargo, el trabajo se ha centrado únicamente en las S-Box que implementa, ya que son la parte crítica y a partir de las cuales se puede desarrollar el conjunto [11].

En concreto estas S-Box utilizan 4 bits de entrada y 4 bits de salida, los cuales están determinados por el circuito que aparece en la Ilustración 22, cuyas ecuaciones se presentan a continuación.

$$A = d\ xor\ (a\ nor\ b)$$

$$B = a\ xor\ (b\ nor\ c)$$

$$C = b\ xnor\ (A\ nor\ c)$$

$$D = c\ xor\ (B\ nor\ A)$$

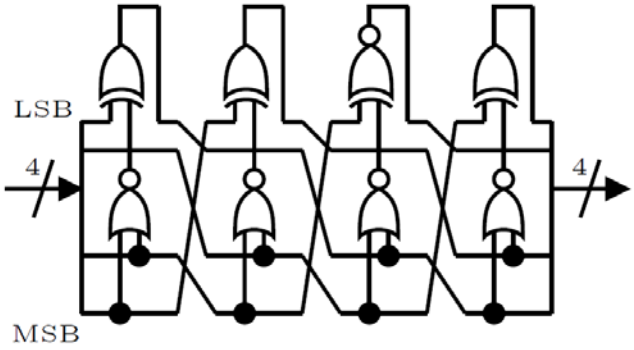


Ilustración 22. S-Box 4 Piccolo [11].

La misión de la Sbox es romper la relación de señales de entrada/salida, de modo que el texto plano se ve modificado a un texto clave en función de una tabla de verdad en formato hexadecimal, la cual se representa en la Tabla 3.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S(x)	e	4	b	2	3	8	0	9	1	a	7	f	6	c	5	d

Tabla 3. Tabla de verdad S-Box 4 Piccolo.

Para realizar la S-Box se ha decidido utilizar lógica WDDL, debido a varios factores, analizados en el Capítulo 3. El primero de ellos es que se ha buscado un diseño que no fuera full-custom, a modo de simplificar la tarea de realizar las celdas y poder centrarnos en lo que es la técnica de layout. Además, otro factor importante ha sido la comparativa frente a las otras lógicas, la cual nos establecía que era una lógica sencilla, y con una relación de área bastante menor que el resto, por lo que, a pesar de sus desventajas, se ha considerado la más recomendable para este estudio.

5.2 Implementación S-Box 4 Piccolo en WDDL

A la hora de implementar esta S-Box, hemos visto en la Ilustración 22 que se utilizan puertas NOR, XOR y XNOR. Esto no es posible aplicarlo a lógica WDDL, ya que únicamente permite puertas AND/NAND y OR/NOR, por lo que las puertas XOR/XNOR deberán ser sustituidas. Al ser una lógica de doble rail, hay que tener en cuenta el diseño de la lógica que genera las señales complementarias y como sería la implementación total de este circuito en lógica WDDL, por lo que, para ello, es necesario conocer las técnicas para la realización de puertas en WDDL. Así, para implementar puertas AND/NAND como OR/NOR en lógica de doble rail se realiza como aparece en la Ilustración 23, mientras que la implementación de puertas XOR/XNOR se realiza como se muestra en la Ilustración 24.

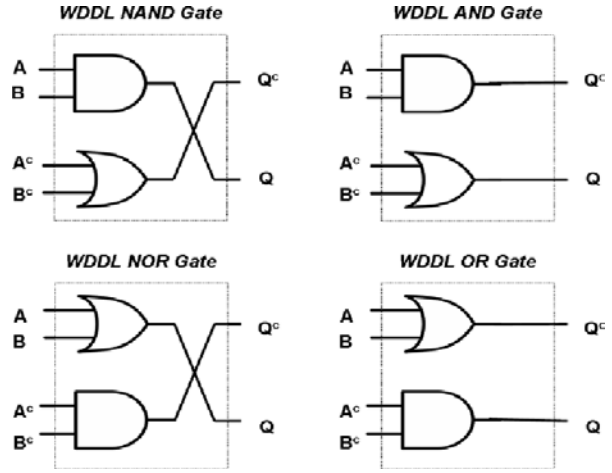


Ilustración 23. Implementación AND/NAND y OR/NOR en WDDL [20]

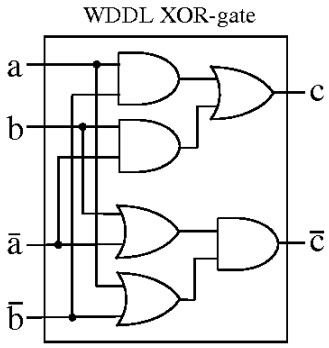


Ilustración 24. Implementación XOR/XNOR en WDDL [21]

5.2.1 Implementación en Verilog

La descripción de este circuito con lógica WDDL para la realización del layout se implementará mediante Verilog en el programa Modelsim que se proporciona desde el Instituto de Microelectrónica de Sevilla y a través del cual se tiene acceso mediante la plataforma RCADs. Así, se ha diseñado un código que respete todas las características necesarias y que se encuentra en el Anexo 1. Este incluye tanto la fase de precarga, como la evaluación, realizando la simulación con una frecuencia del reloj de 100MHz, frecuencia que se puede considerar como adecuada para este tipo de aplicaciones. La precarga se realizará con el reloj en nivel alto, mientras que la evaluación se realiza cuando el reloj está en nivel bajo, obedeciendo al esquema que se muestra en la Ilustración 25.

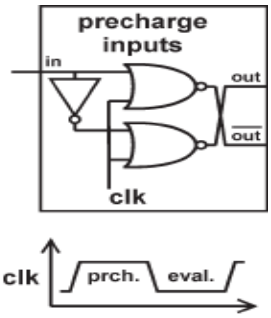


Ilustración 25. Fase de precarga en WDDL [16]

Para generar las señales durante la fase de precarga se ha implementado el circuito de la Ilustración 25. Con esto lo que se consigue es que estando el reloj a 1, todas las señales de entrada a la S-box sean 0. Según la funcionalidad del código, todas las señales del circuito, incluidas las salidas, serán 0. (Recuadro amarillo de la Ilustración 26).

Para la fase de evaluación se utiliza el mismo el circuito de la Ilustración 25. Con esto, cuando el reloj está a 0 se realiza la lógica que proporciona la transformación de texto plano a texto cifrado. Para ello, se ha tenido en cuenta que es necesario realizar un único cambio de valores de datos de entrada para cada evaluación. (Recuadro azul de la Ilustración 26).

Posteriormente se ha realizado un testbench para probar el correcto funcionamiento y se obtuvieron los resultados que se aprecian en la Ilustración 26. Este testbench se encuentra incluido en el anexo 2.

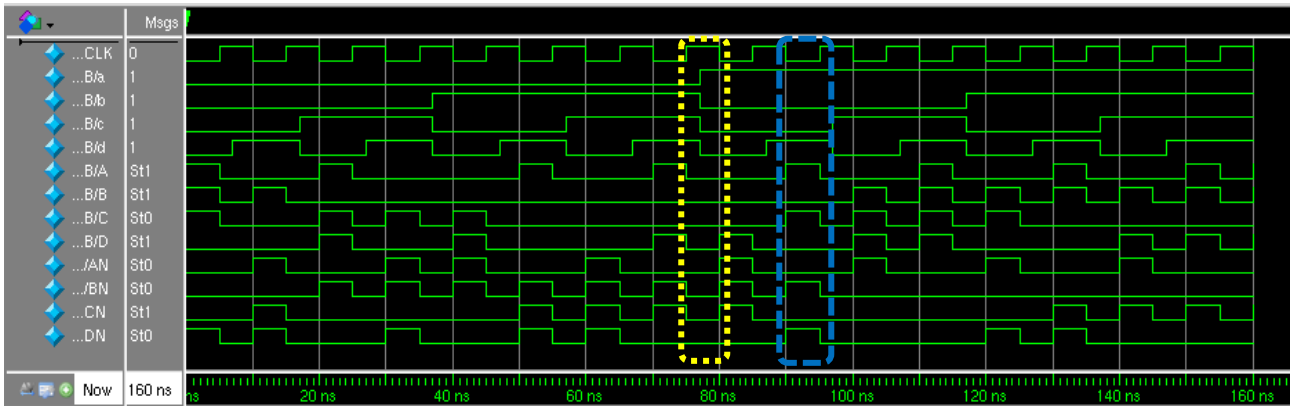


Ilustración 26. Simulación S-Box 4 Piccolo.

Si se observa la simulación de la Ilustración 26, puede verse como en cada señal de precarga (ejemplo del recuadro amarillo), las salidas son 0 aunque obliguemos un valor en las entradas. Esto está hecho a propósito, para que cuando llegue el momento de la evaluación, las entradas ya se encuentren estables.

Los resultados obtenidos pueden analizarse uno por uno, y se puede comprobar como para cada una de las entradas, se obtienen las salidas binarias correspondientes a la Tabla 4.

a	b	c	d	A	B	C	D
0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	1
0	1	0	1	1	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	1	1	1
1	1	0	0	0	1	1	0
1	1	0	1	1	1	0	0
1	1	1	0	0	1	0	1
1	1	1	1	1	1	0	1

Tabla 4. Resultado simulación S-Box 4 Piccolo.

Si ahora se realiza la comparación de la Tabla 4 (valores binarios), con la Tabla 3 (valores en hexadecimal), se puede comprobar que los resultados obtenidos son los correctos.

5.2.2 Síntesis lógica

Tras haber realizado el código y comprobar su correcto funcionamiento con el testbench, es hora de realizar la síntesis lógica y aplicar las restricciones que queramos; en este caso serán restricciones temporales con un reloj de 100MHz. Para ello se ha utilizado la herramienta que se proporciona en RCADs para realizar estas tareas llamada Design Vision, utilizando la tecnología UMC180 que es la que se ha trabajado en el máster y con la que se ha realizado este trabajo al completo.

Una vez nos situamos en esta herramienta y cargamos el código Verilog generado, si vemos el circuito como un esquemático obtenemos lo que se muestra en la Ilustración 27.

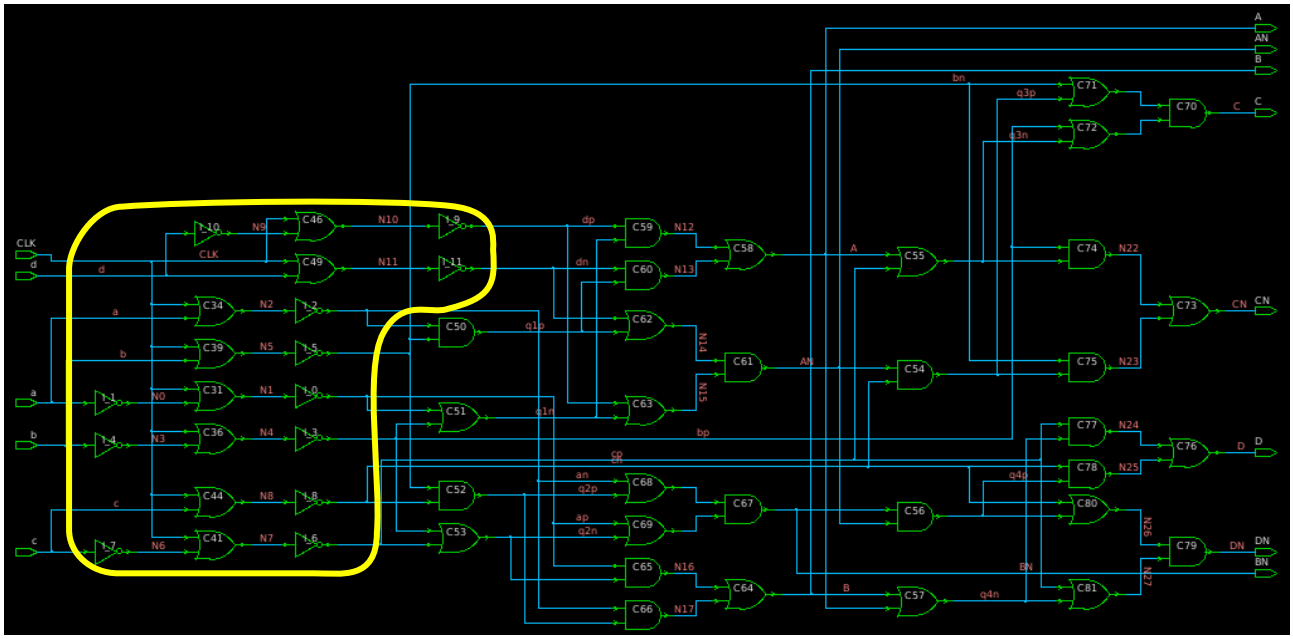


Ilustración 27. Esquemático S-Box 4 Piccolo sin sintetizar.

En la zona izquierda de la Ilustración 27 dentro de un recuadro amarillo, se puede observar el circuito encargado de generar la precarga, compuesto por puertas OR e inversores. Concretamente, se ha decidido poner estos inversores frente a la posibilidad de intercambiar la señal original y la complementaria, como ocurre en la Ilustración 13, para mantener la originalidad del circuito de precarga de la Ilustración 25. Como este circuito no entra dentro de lo que sería la lógica combinacional de la S-Box (Ilustración 22), las puertas utilizadas para la lógica WDDL seguirían siendo positivas, algo imprescindible para evitar glitches como se menciona en el capítulo 3.

En la zona derecha, fuera del recuadro amarillo, se encuentra toda la lógica combinacional perteneciente a la S-Box 4 Piccolo, formada únicamente por puertas AND y OR.

Tras la realización de la síntesis, el circuito resultante es el que se presenta en la Ilustración 28.

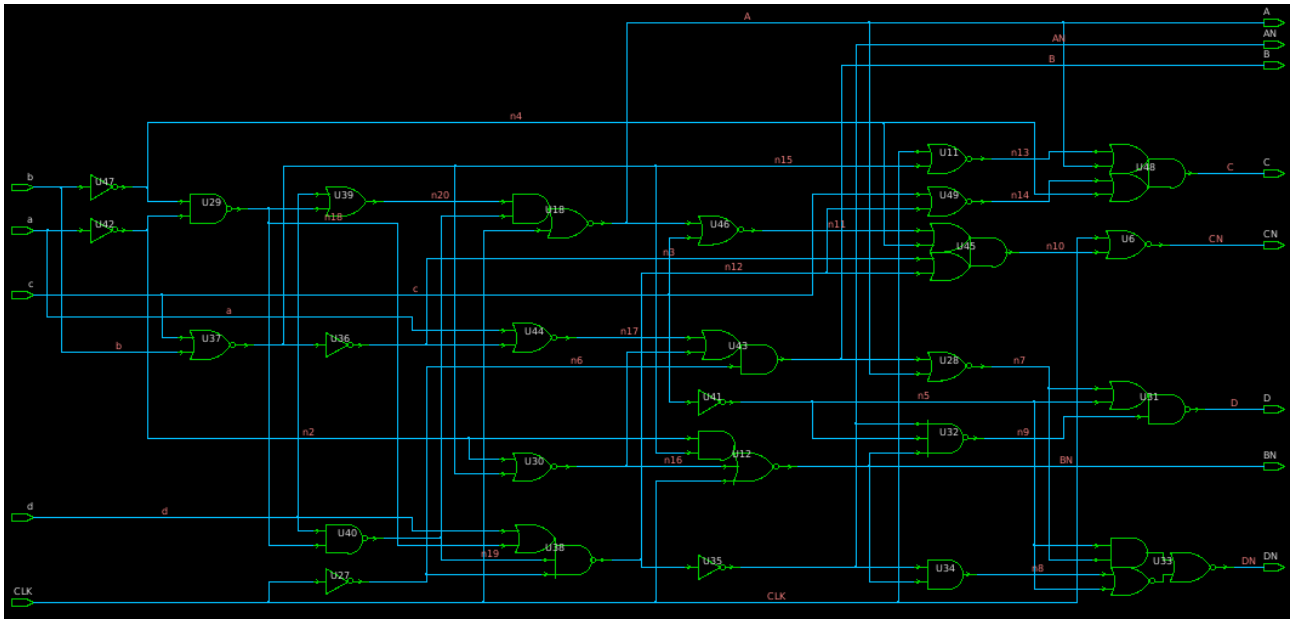


Ilustración 28. Esquemático S-Box 4 Piccolo sintetizado.

En este circuito obtenido tras realizar la síntesis, a primera vista puede llamar la atención la existencia de inversores no permitidos en lógica WDDL, pero es necesario fijarse que la lógica combinacional de la operación es positiva, ya que la puerta lógica anterior o posterior del inversor está negada.

Del circuito sintetizado se obtiene un diseño cuyos parámetros quedan resumidos en la Tabla 5

N.º Celdas	Pines entrada	Pines salida	Área de celdas [um^2]	Consumo dinámico [uW]	Path Slack [ns]
27	5	8	111	66,05	c-DN [0.86]

Tabla 5. Parámetros circuito sintetizado.

Según el path slack obtenido, la frecuencia máxima a la que podría funcionar el circuito al realizar la síntesis es $f = \frac{1}{T} = \frac{1}{0.86\,ns} = 1.1628\,GHz$. Sin embargo, es necesario tener en cuenta que esta frecuencia está calculada en base únicamente a las celdas, sin tener en cuenta las interconexiones entre celdas, impedancias o capacitancias entre ellas, por ejemplo, por lo que esto sería una frecuencia máxima con parámetros ideales, algo inalcanzable.

Para comprobar el correcto funcionamiento del circuito sintetizado y realizar la verificación temporal, se ha obtenido el código en Verilog y, además, el fichero SDF, que es el que incluye toda la información sobre retrasos. Esto se realiza para ver cómo influyen los retrasos en las puertas y ver que, a pesar de estos, no existe ningún problema.

5.2.3 Simulación post síntesis

Tras realizar la síntesis, es necesario obtener una simulación con el nuevo circuito, así como el fichero que contiene la información de los retrasos, lo cual se ha realizado con Modelsim.

Una vez se tienen estos ficheros disponibles y realizamos la simulación, se obtienen los resultados mostrados en la Ilustración 29, Ilustración 30 e Ilustración 31.

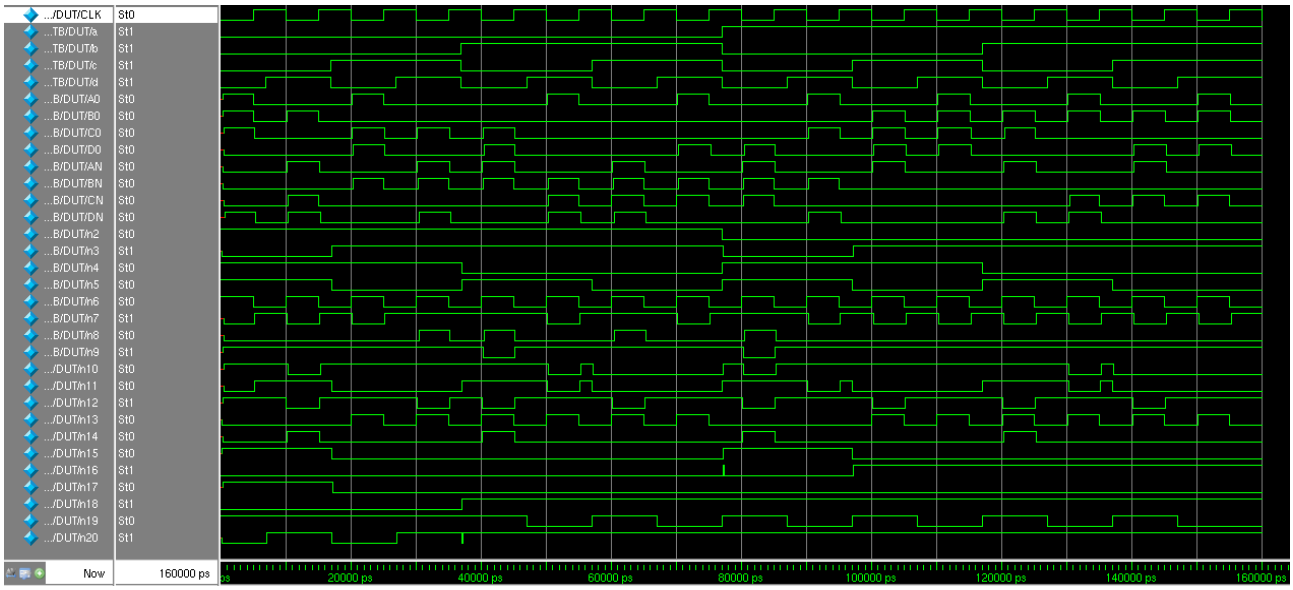


Ilustración 29. Simulación post síntesis.

Si se analiza la Ilustración 29, se puede comprobar que el valor de las salidas es correcto respecto a las entradas del circuito y cumple con la tabla de verdad que se muestra en la Tabla 3. Si se hace zoom en algunas zonas, se puede ver más claramente cómo existen glitches que no influyen en el correcto funcionamiento del circuito incluso a mayores frecuencias de operación. Estos retrasos se pueden observar más detalladamente en la Ilustración 30 e Ilustración 31 y, son del orden de los 500 ps entre que se activa la señal de reloj y se produce el cambio en la última salida del circuito.

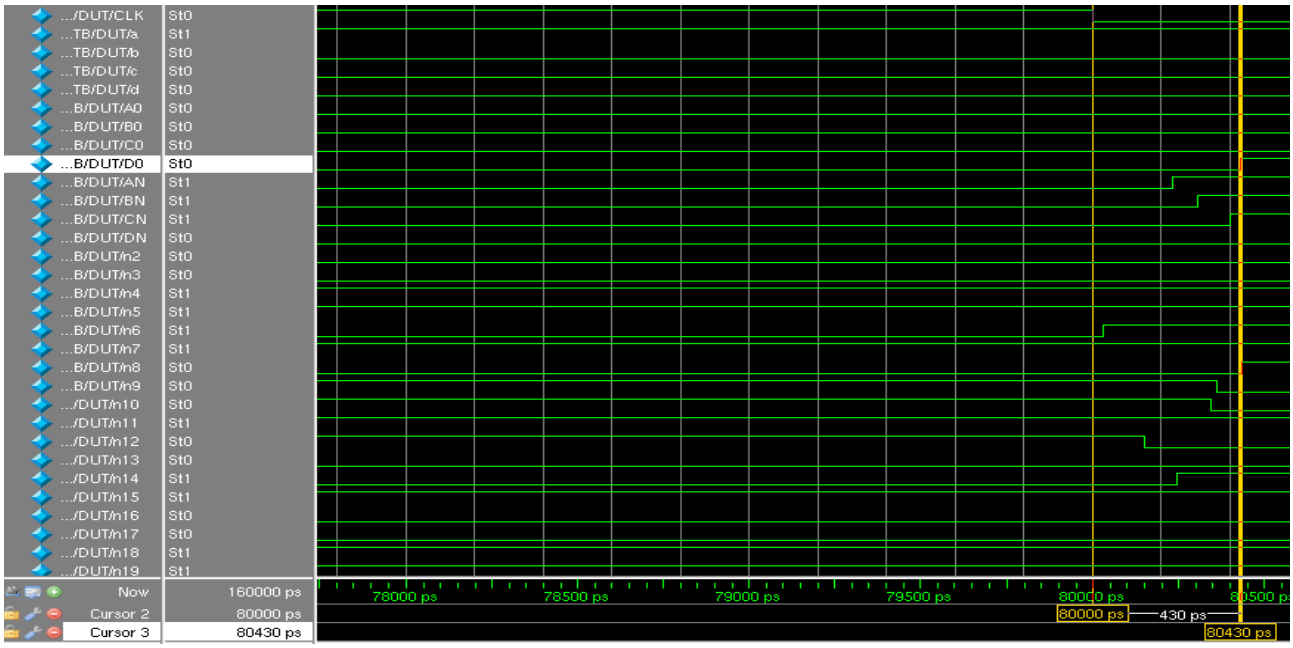


Ilustración 30. Simulación post síntesis zoom 1.

En la Ilustración 30 se observa un retraso de 430 ps entre que se produce el flanco de reloj que activa la evaluación y el último cambio en la salida D0.

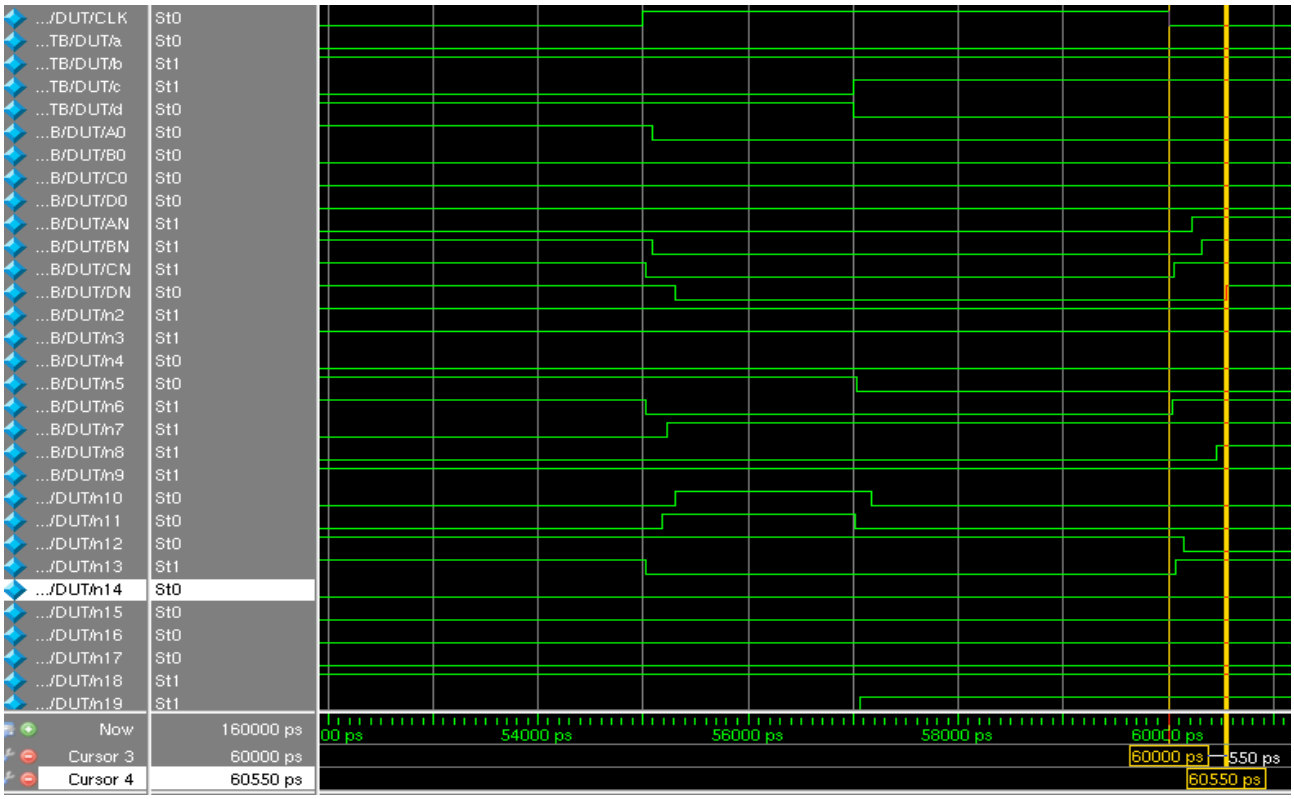


Ilustración 31. Simulación post síntesis zoom 2.

Algo similar ocurre en la Ilustración 31, que muestra un retraso de 550 ps entre que se produce el flanco de reloj que activa la evaluación y el último cambio en la salida DN. De este modo, se ve claramente como las entradas que modifican DN, hacen que el retraso máximo aumente, confirmando lo obtenido en la síntesis, que el camino crítico es el existente entre la entrada c y la salida DN.

Esta magnitud de retrasos está generada debido a las puertas lógicas, ya que es un circuito que además de no estar optimizado para alta velocidad, tiene una considerable profundidad lógica. Sin embargo, esto no es un problema, ya que las aplicaciones criptográficas no suelen estar diseñadas para operar a alta frecuencia.

5.2.4 Layout

Una vez se ha realizado la simulación post síntesis y se ha comprobado el correcto funcionamiento, se ha procedido a efectuar el layout del diseño. Para mantener la utilidad de la herramienta RCADs y los medios disponibles en el IMSE, se ha hecho el layout utilizando el programa INNOVUS.

Para ello se han seleccionado unos parámetros estándar como los que se muestran en uno de los tutoriales de INNOVUS proporcionados para la asignatura de Diseño y Metodologías de herramientas CAD. Este tutorial en concreto se encuentra en la biografía con la referencia [22]. Estos parámetros que tomaremos como estándar hacen referencia principalmente al floorplan, que se mantiene como el programa lo deja por defecto, con una utilización del 84% y una ratio de aspecto de 0.55. Los anillos de masa y alimentación con un espesor de 2 um y un offset de 5 um. Se incluyen fillers de 1, 2, 4, 8, 16, 32 y 64 y respecto al layout las opciones por defecto, activando las opciones Timing Driven y SI Driven, dejando el esfuerzo en 5. Al realizar todo este procedimiento, el layout resultante es el que se muestra en la Ilustración 32.

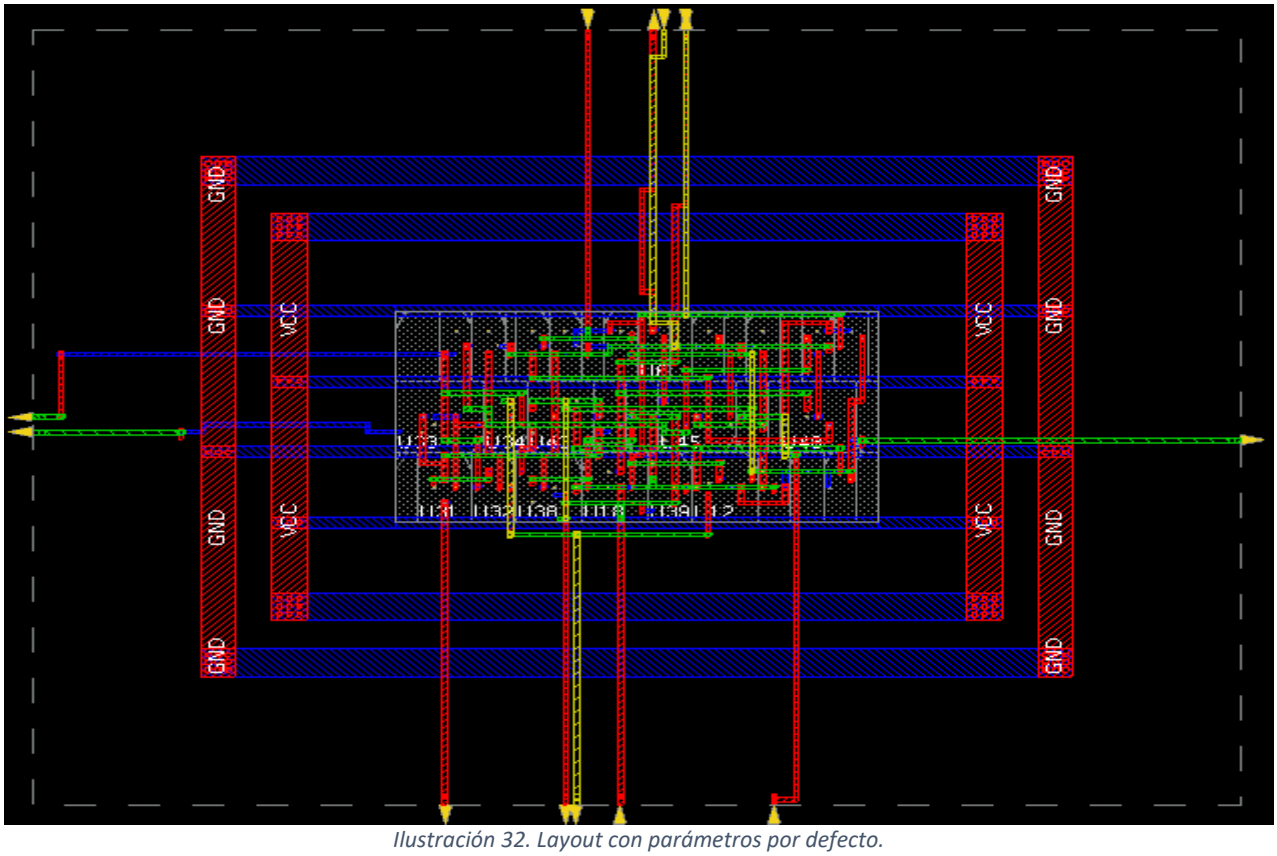


Ilustración 32. Layout con parámetros por defecto.

En el layout podemos ver cómo el resultado es un diseño con 3 filas de celdas, varios fillers, y la disposición de los pines de entrada/salida ha sido designada automáticamente por el programa.

Como se ha comentado en el comienzo del Capítulo 5, se quiere trabajar con un diseño orientado para *Fat Wire*. Para ello lo que se busca es una máxima simetría entre cada señal de salida y su complementada, por lo que se ha actuado sobre la posición de los pines de salida, es decir, A0 y AN,

B0 y BN, C0 y CN y D0 y DN. El grado de simetría conseguido se ha medido a través del producto RC de los nodos de estas señales, de modo que a mayor igualdad entre ellos mayor simetría habremos obtenido.

Los datos empleados para el cálculo de simetría han sido el valor de resistencia y capacitancia de las señales de salida. Para ello se ha extraído la información mediante el comando *Extract RC* de INNOVUS, seleccionando las opciones *set load* y *set resistance* para obtener la capacitancia global y resistencia respectivamente por nodo. Una vez se ha dispuesto de esta información, ha sido trasladada a Excel y mediante una macro se ha realizado un procesado de los datos, obteniendo para todos los nodos el valor del producto RC de cada señal y su complementada.

Tras disponer del producto RC de las salidas que nos interesa se ha hecho una comparación entre cada señal de salida y su complementada, calculando la variación relativa de la señal complementada respecto de la original.

Posteriormente, se ha calculado la media de las variaciones del producto RC entre los cuatro nodos de salida y sus complementadas, siendo este el valor mediante el cual se han comparado las propuestas.

Todo el contenido referente a estos datos de capacitancia, resistencia, producto RC y variaciones de los nodos para cada ensaño se encuentra incluido en el Anexo 4, por lo que en el capítulo se han empleado únicamente los resultados finales.

Concretamente para este diseño obtenido con los parámetros por defecto, se han obtenido unos valores de variación para el producto RC entre los nodos del 1.39% para el nodo A, 56.23% para el nodo B, 38.89% para el nodo C y 72.73% para el nodo D, lo que hace una variación media del 42.31%.

Se ha realizado un resumen gráfico de este bloque en la Ilustración 33, así como un resumen de los parámetros más importantes en la Tabla 6.

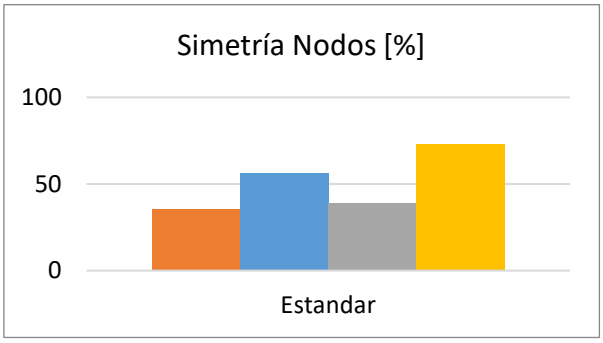


Ilustración 33. Gráfica simetría ensayo estándar.

Ensayo	Área total celdas [um^2]	Área total chip [um^2]	N.º Filas	Espaciado filas[um]	Ratio aspecto	Fillers añadidos	Variación media [%]
0	412.474	3781.008	3	0	0.554	3x1, 7x2, 1x4	42.31

Tabla 6. Resumen parámetros diseño estándar.

6 COMPARACIÓN DE LAYOUTS Y SIMETRÍAS EN PINES CON MODIFICACIÓN DE PARÁMETROS

Tras haber realizado un diseño con parámetros por defecto y tener una base de partida, la idea es seguir diferentes estrategias para obtener diferentes layouts frente a un mismo circuito lógico sintetizado con Design Vision. Con esto conseguiremos ver cuáles son los parámetros que más afectan a la hora de diseñar un layout enfocado para *Fat Wire*.

Así, se han realizado varios ensayos para verificar la influencia de los diferentes parámetros en un layout, comparando el producto de las resistencias con las capacidades en los nodos de salida para todas las señales diferenciales en cada uno de los diferentes ensayos realizados.

Para la realización de cada ensayo se ha modificado únicamente un parámetro seleccionado cada vez y el resto se han mantenido iguales al del diseño por defecto, mientras que para la comparativa que posteriormente se ha realizado y en la que se resumen todos los valores, se ha tenido en cuenta la variación media de los porcentajes de simetría (valores RC) de los nodos de salida.

Concretamente, los parámetros que se han modificado se mencionan a continuación y se detalla lo esperado.

- Establecer un número determinado de filas de celdas. Con esto se pretende encontrar cual es el número de filas más adecuado para este diseño en concreto. El número de filas de celdas con el que cuenta el floorplan se espera que influya en el resultado final, pero no es un parámetro con el que pueda generalizarse que el efecto obtenido sea igual en todos los diseños, ya que, dependiendo de la complejidad del diseño del que se trate, el número óptimo de filas variará. Para establecer el número de filas se ha modificado la ratio altura/anchura (H/W) en INNOVUS, recuadro de amarillo de la Ilustración 34.

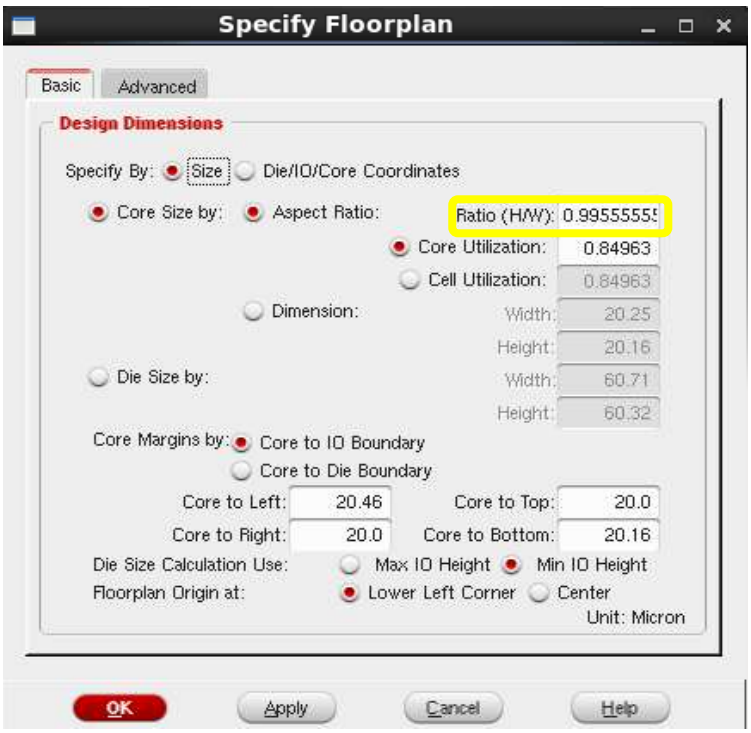


Ilustración 34. Modificación ratio aspecto.

- Establecer una separación entre filas. Este parámetro se espera que sea muy influyente, ya que, al separar más las filas del posicionado de celdas, se pretende una menor congestión de routing, lo que mejorará la simetría entre las señales reduciendo el cruce de ellas. Esto se selecciona en el recuadro amarillo de la Ilustración 35. Variar el posicionado de alimentación y masa. Al igual que ocurre con el número de filas, se espera que sea un parámetro que varíe el resultado final, pero del cual no se puede generalizar, ya que dependerá del diseño en concreto con el que se trabaja. Dicho parámetro se modifica en la opción en rojo de la Ilustración 35.

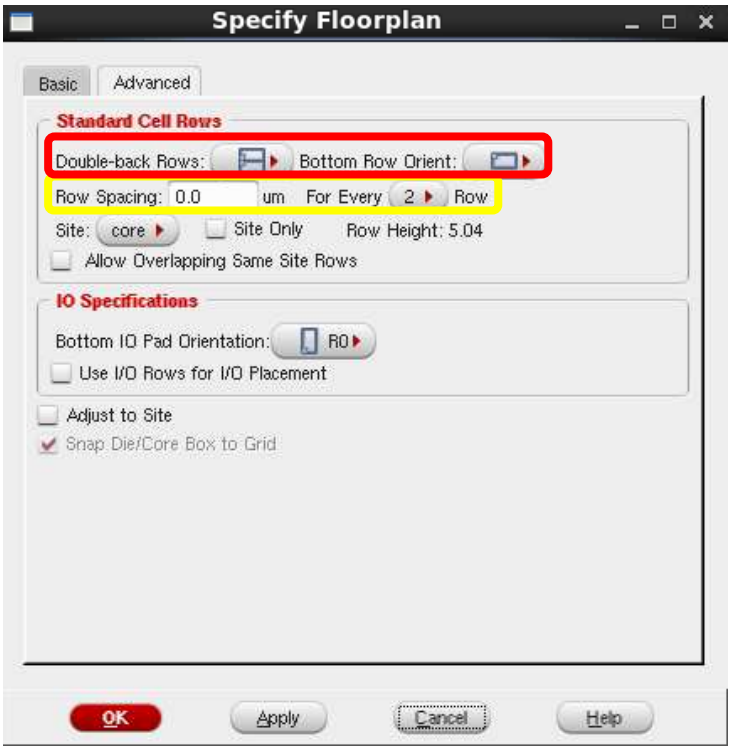


Ilustración 35. Modificación del espaciado de filas.

- Activar parámetros de ayuda a mejora como SI Driven o el esfuerzo de routing. En este caso se espera que los resultados mejoren al activar estos parámetros respecto de si no se hace, ya que se tratan como una ayuda para la optimización. Esta activación o desactivación de parámetros se realiza seleccionando las opciones del recuadro amarillo de la Ilustración 36, donde además de activarlos, se puede seleccionar el esfuerzo en unos niveles que van desde el 0 al 10.

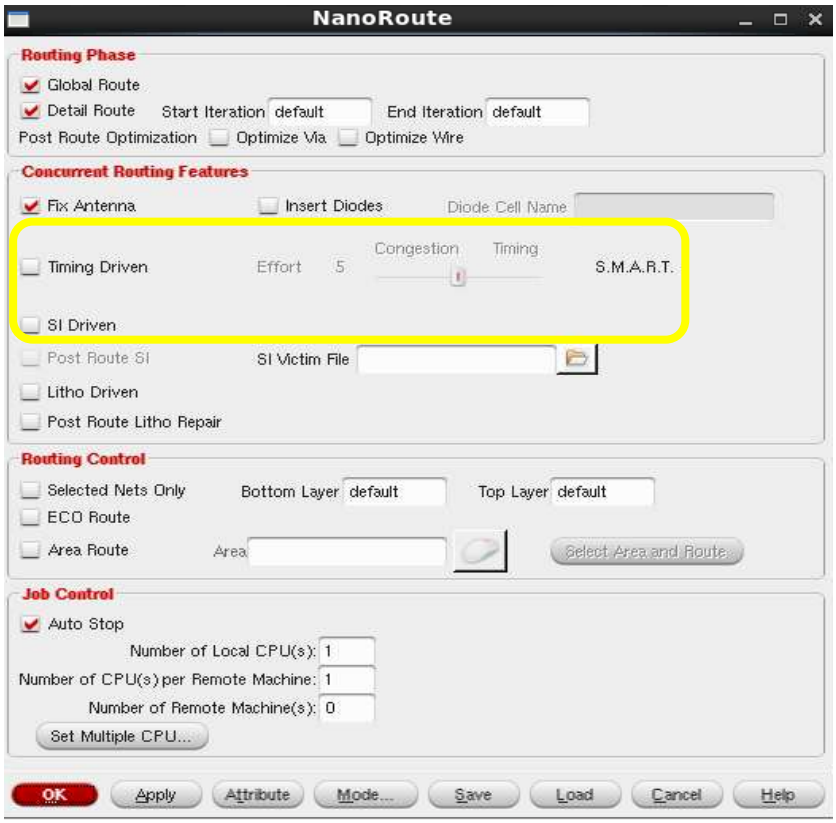


Ilustración 36. Parámetros de ayuda y esfuerzo.

- Posicionar celdas y pines manualmente. Definitivamente, este es el parámetro del que más se espera, ya que se obliga a colocar celdas y pines simétricamente. El inconveniente de esto es que ya se elimina la automatización del proceso, pero a cambio, se esperan unos resultados considerablemente positivos. Para realizar esta opción es necesario cargar justo antes de hacer el layout un fichero con extensión *.io.* El contenido del fichero utilizado para obtener los resultados que se mostrarán a continuación se encuentra en el anexo 3.

Una vez se ha realizado la modificación de cada bloque y se han extraído los datos, se ha realizado una tabla resumen donde se incluyen los principales parámetros del diseño, y se puede encontrar al final de cada apartado.

6.1 Número de filas en el floorplan

Para realizar este conjunto de ensayos, se han realizado 4 diseños, en el que se comparan alternativas con 4, 3, 2 y 1 filas (ensayo del 1 al 4 respectivamente). Para ello se ha ido modificando la ratio de aspecto de modo que pudiéramos controlar la cantidad de filas que tenía el diseño. Concretamente para establecer 4 filas en el floorplan, se ha requerido una ratio de aspecto de 0.985.

Tras realizar el diseño para el ensayo 1 se han obtenido unos valores de variación para el producto RC entre los nodos del 35.51% para el nodo A, 188% para el nodo B, 0% para el nodo C y 215.15% para el nodo D, lo que hace una variación media del 109.66%.

Para poder realizar un floorplan con 3 filas se ha establecido la ratio de aspecto a 0.554, obteniendo para el ensayo 2 unos valores de variación para el producto RC entre los nodos del 1.39% para el nodo A, 56.23% para el nodo B, 38.89% para el nodo C y 72.73% para el nodo D, lo que hace una variación media del 42.31%.

Con el fin de tener 2 filas para el posicionado de celdas (ensayo 3), se ha seleccionado una ratio de aspecto de 0.246 obteniendo unos valores de variación para el producto RC entre los nodos del 66.5% para el nodo A, 101.55% para el nodo B, 9.09% para el nodo C y 9.09% para el nodo D, lo que hace una variación media del 46.56%.

Para el último ensayo relacionado con la ratio de aspecto (ensayo 4), este se ha establecido en 0.061, obteniendo así un diseño de una única fila, y consiguiente unos valores de variación para el producto RC entre los nodos del 81.25% para el nodo A, 629.82% para el nodo B, 0% para el nodo C y 0% para el nodo D, lo que hace una variación media del 177.77%.

Se ha realizado un resumen gráfico de este bloque en la Ilustración 37, así como un resumen de los parámetros más importantes en la Tabla 7. En esta gráfica podemos ver que tanto para el ensayo 1(4 filas) como para el ensayo 4(1 filas) se obtienen unos niveles de asimetría muy altos, por lo que se consideran opciones pésimas para este diseño. Sin embargo, en los ensayos 2 y 3 (3 y 2 filas respectivamente), se observa cómo hay un nivel de simetría generalizado, aunque se obtiene un mejor resultado en el ensayo 2 (3 filas).

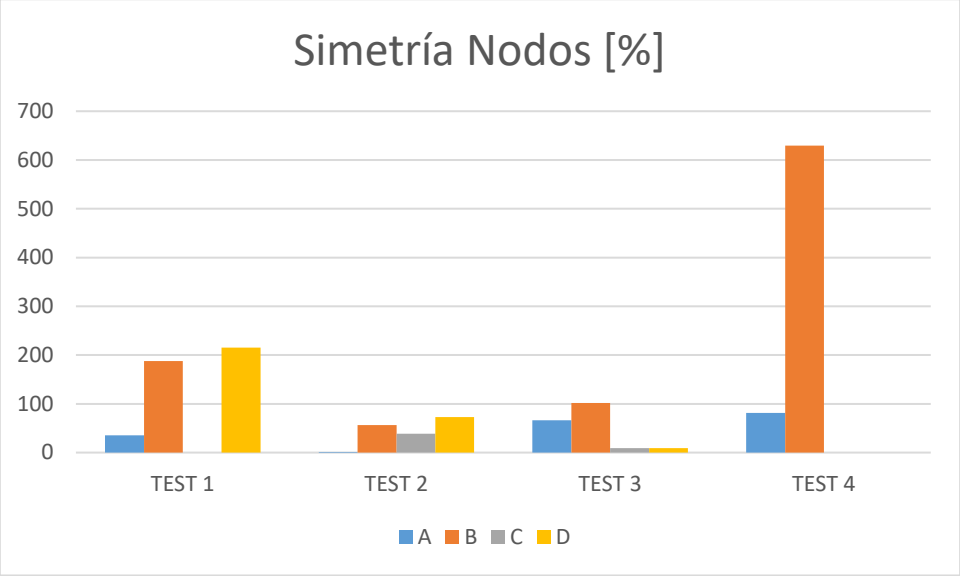


Ilustración 37. Gráfica simetría ensayo nº filas.

Ensayo	Área total celdas [um^2]	Área total chip [um^2]	N.º Filas	Espaciado filas[um]	Ratio aspecto	Fillers añadidos	Variación media [%]
1	412.474	3712.262	4	0	0.985	5x1,2x8	109.66
2	412.474	3781.008	3	0	0.554	3x1, 7x2, 1x4	42.31
3	412.474	4124.736	2	0	0.246	7x1, 3x2, 2x4	46.56
4	412.474	5568.394	1	0	0.061	5x1,2x2,1x4,1x8	177.77

Tabla 7. Resumen parámetros modificación filas.

6.2 Variación del espaciado de filas

Para este bloque de casos se ha tratado de modificar el espaciado existente entre cada fila de celdas a la hora de realizar el floorplan con valores de: 1.12 um, 2.24 um, 3.36 um, 4.48 um y 5.56 um.

Para el ensayo 5 se ha establecido un espaciado de 1.12 um y se ha obtenido así un diseño como el que se muestra en la Ilustración 38. Los valores de variación para el producto RC entre los nodos son del 38.46% para el nodo A, 461.84% para el nodo B, 31.25% para el nodo C y 215.15% para el nodo D, lo que hace una variación media del 186.68%.

Como se observa en la Ilustración 38, los pines de entrada/salida de datos están colocados en diferente posición respecto al ensayo estándar, por lo que este diseño y sus homólogos en separación de filas, únicamente pueden ser comparados entre ellos. A priori no se ha seleccionado nada sobre el posicionado de pines y se ha dejado libertad a lo que el programa decide automáticamente, por lo que este parámetro no se tiene en cuenta, hasta un proceso que posteriormente se verá.

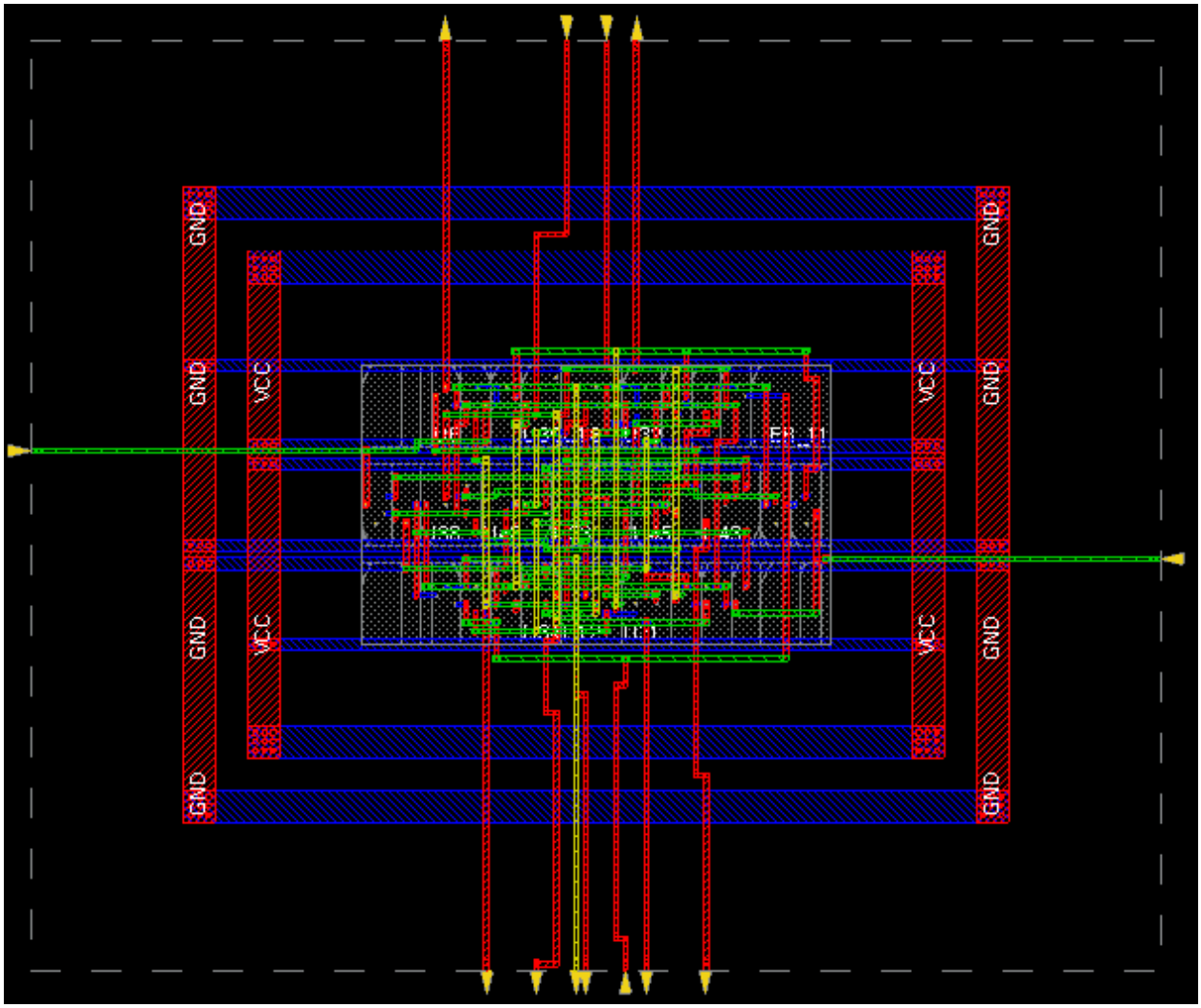


Ilustración 38. Diseño con espaciado de filas a 1.12 um.

Siguiendo el mismo protocolo que para el anterior, pero aumentando la separación entre filas a 2.24 um (ensayo 6), se obtienen como resultados finales que la variación para el producto RC entre los nodos es del 38.18% para el nodo A, 334% para el nodo B, 36.54% para el nodo C y 215.15% para el nodo D, lo que hace una variación media del 155.97%.

Para el ensayo 7, en el que se aumenta la separación entre filas a 3.36 um, el resultado muestra una variación para el producto RC entre los nodos del 23.29% para el nodo A, 281.82% para el nodo B, 78.85% para el nodo C y 38.67% para el nodo D, lo que hace una variación media del 105.65%.

Continuando con el aumento de separación de filas, se alcanza una distancia de 4.48 um en el ensayo 8, obteniendo una variación para el producto RC entre los nodos del 11.62% para el nodo A, 11.98% para el nodo B, 67% para el nodo C y 100% para el nodo D, lo que hace una variación media del 47,65%.

Finalmente se ha alcanzado el espaciado entre filas de 5.56 um para el ensayo 9, con unos resultados de una variación para el producto RC entre los nodos del 77.78% para el nodo A, 43.09% para el nodo B, 11.11% para el nodo C y 8.33% para el nodo D, lo que hace una variación media del 35.08%.

Se ha realizado un resumen gráfico de este bloque en la Ilustración 39, así como un resumen de los parámetros más importantes en la Tabla 8. En este, podemos observar como a medida que se va aumentando el espaciado entre filas, se va consiguiendo una mejor simetría, reduciendo la disparidad entre el producto RC de los diversos nodos. Por consiguiente, a mayor espaciado, más óptimo será el diseño en lo que a simetría de los nodos de salida respecta.

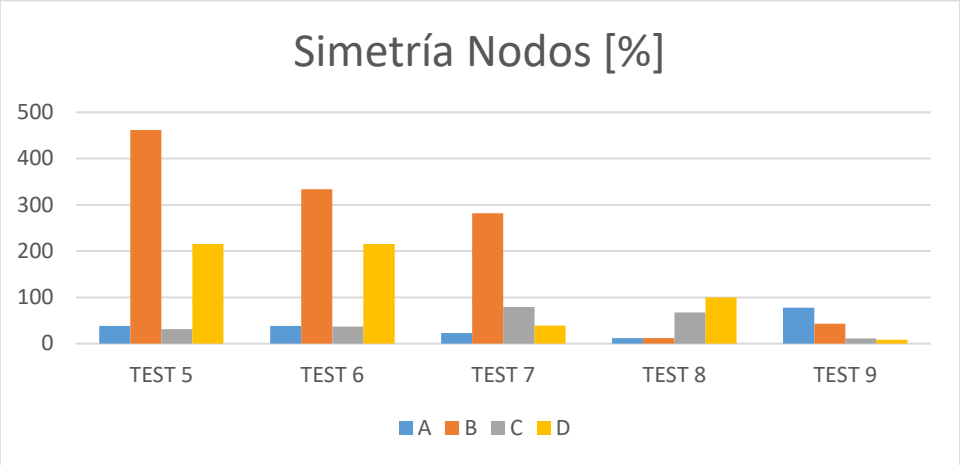


Ilustración 39. Gráfica simetría ensayo espaciado.

Ensayo	Área total celdas [um^2]	Área total chip [um^2]	N.º Filas	Espaciado filas[um]	Ratio aspecto	Fillers añadidos	Variación media [%]
5	440.597	4041.061	3	1.12	0.596	4x1,3x2,3x4 ,1x8	186.68
6	459.346	4272.296	3	2.24	0.645	2x1,1x2,4x8	155.97
7	478.094	4509.086	3	3.36	0.691	2x1, 2x2, 1x4, 4x8	105.65
8	487.469	4711.504	3	4.48	0.747	3x1, 5x2, 4x8	47.65
9	496.843	4916.699	3	5.6	0.801	6x1, x 9x2, 4x4, 1x8	35.08

Tabla 8. Resumen parámetros modificación espaciado.

6.3 Variación de posicionado relativo de alimentación y masa

Para la realización de este grupo de ensayos se ha variado el parámetro Double-back Rows de lo que viene por defecto, que es bottom-to-bottom y se ha establecido como top-to-bottom. Además, como era necesario poner un espaciado entre filas para que no hubiera cortocircuito entre las líneas de masa y alimentación por el hecho de estar solapadas, se ha seleccionado un espaciado anteriormente realizado en otros ensayos para poder comparar, al igual que con el diseño del siguiente punto. En este caso se ha seleccionado una distancia de 2.24 um, y de 4.48 um, haciéndolos comparables con diseños anteriores.

Para poder mostrar el cambio realizado, se ha incluido la Ilustración 40, en la que se muestra que además del espaciado, las líneas de masa y alimentación han cambiado su distribución respecto a la Ilustración 38, por ejemplo.

En primera instancia además de aplicar el posicionado, se ha establecido para el ensayo 10 una separación de 2.24 um entre filas, obteniendo unos resultados que corresponden a una variación para el producto RC entre los nodos del 42.86% para el nodo A, 140% para el nodo B, 36.54% para el nodo C y 188.89% para el nodo D, lo que hace una variación media del 102.07%.

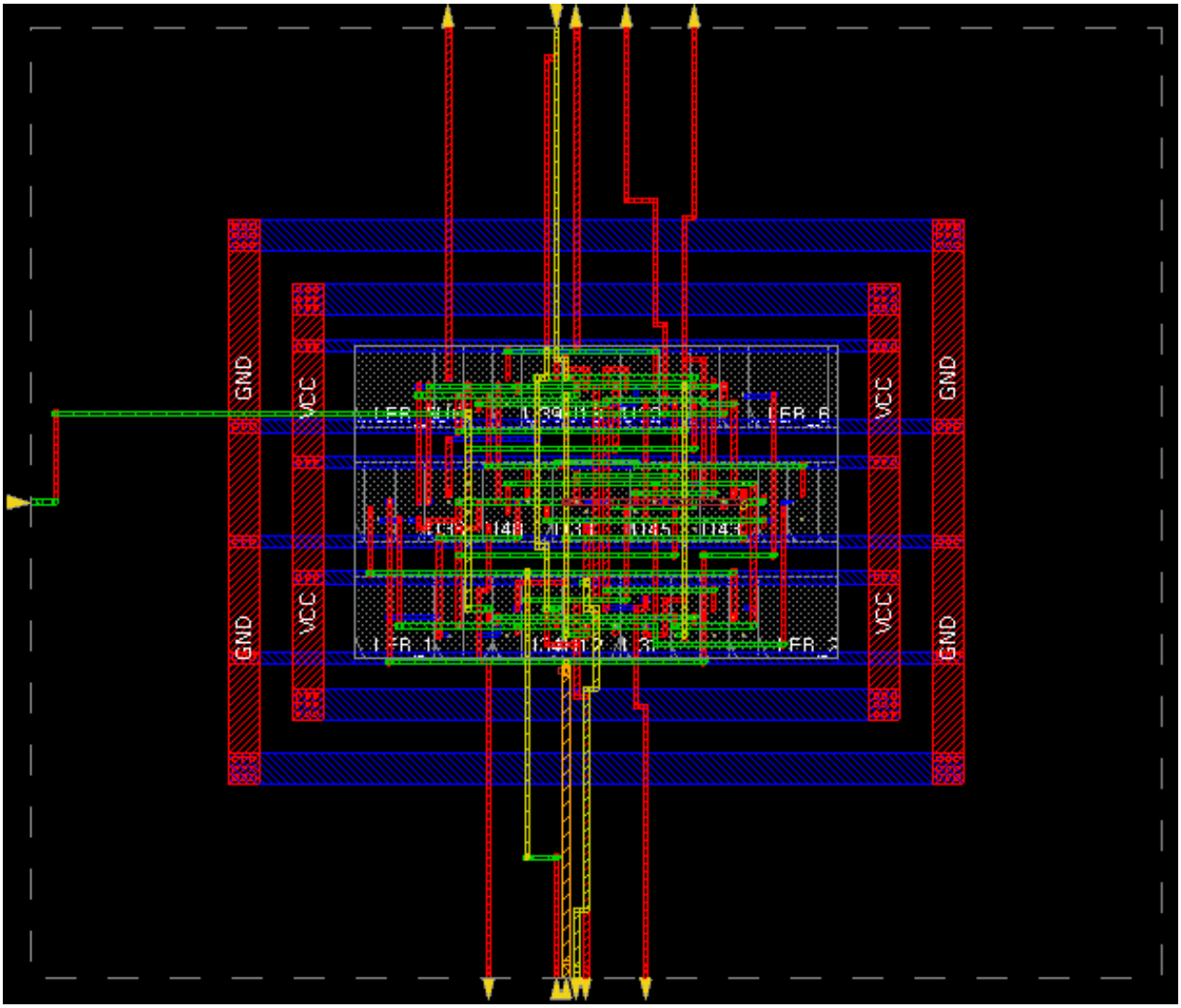


Ilustración 40. Diseño Double-back Rows top-to-bottom y espaciado de filas fijado a 2.24 um.

Para el ensayo 11, en el que se mantienen los mismos parámetros que en el ensayo 10, pero variando la separación entre filas a 4.48 um, los resultados son una variación para el producto RC entre los nodos del 16.25% para el nodo A, 89.29% para el nodo B, 58.75% para el nodo C y 150% para el nodo D, lo que hace una variación media del 78.57%.

Se ha realizado un resumen gráfico de este bloque en la Ilustración 41, así como un resumen de los parámetros más importantes en la Tabla 9, donde de nuevo podemos comprobar que, a mayor espaciado, mayor simetría. Sin embargo, podemos observar como para un espaciado de 2.24 um se obtienen mejores resultados que en el caso del ensayo 6, mientras que para el espaciado de 4.48 um tenemos un peor resultado que en el ensayo 8. Esto ya nos proporciona información muy valiosa, ya que nos muestra que es un parámetro cuya influencia en el resultado final depende del diseño con el que se está trabajando.

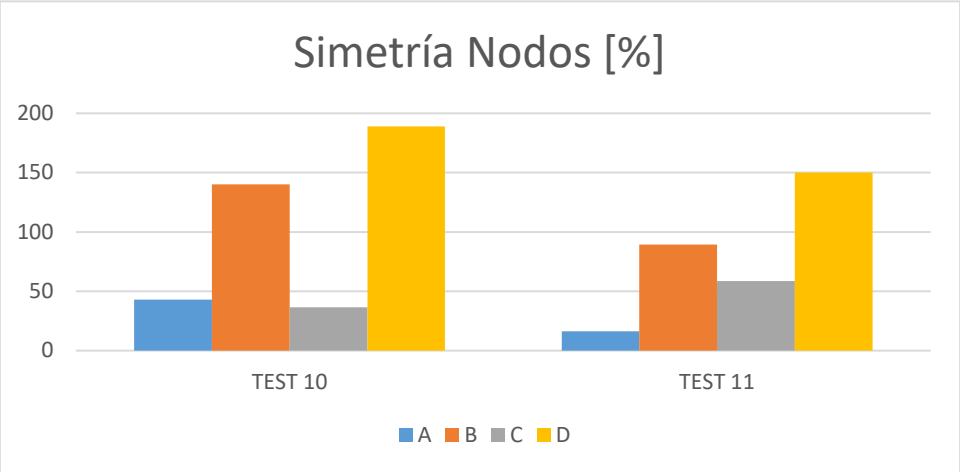


Ilustración 41. Gráfica simetría ensayo cambio GND/VCC y espaciado.

Ensayo	Área total celdas [um^2]	Área total chip [um^2]	N.º Filas	Espaciado filas[um]	Ratio aspecto	Fillers añadidos	Variación media [%]
10	459.346	4272.296	3	2.24	0.645	2x1,1x2,4x8	102.07
11	487.469	4711.504	3	4.48	0.747	3x1, 5x2, 4x8	78.57

Tabla 9. Resumen parámetros modificación posición VCC y GND.

6.4 Activación de parámetros de ayuda

En este caso hay que dividir el proceso en dos diferentes opciones: SI Driven y esfuerzo.

El SI Driven ayuda a prevenir o reducir el acoplo entre señales, mientras que el esfuerzo hace referencia a la agresividad del routing para cumplir con las restricciones temporales.

6.4.1 SI Driven

Para este ensayo se han mantenido todos los parámetros por defecto y se ha jugado con la activación o desactivación de este parámetro.

Si realizamos el mismo procedimiento que para el ensayo 0, pero con el parámetro SI Driven desactivado (ensayo 12), se obtienen unos resultados de variación del producto RC entre nodos de

37.5% para el nodo A, 136.47% para el nodo B, 42.1% para el nodo C y 72.73% para el nodo D, lo que hace una variación media del 72.2%.

Concretamente la activación de este parámetro es lo que viene por defecto, por lo que este ensayo 13 debería proporcionar los mismos resultados que el denominado ensayo estándar o ensayo 0. A pesar de esto, se ha decidido realizarlo, para así comprobar los resultados y que todo el procedimiento es correcto, obteniendo los mismos resultados que para dicho ensayo 0.

Se ha realizado un resumen gráfico de este bloque en la Ilustración 42, así como un resumen de los parámetros más importantes en la Tabla 10, donde se muestra que este parámetro es influyente, concediendo una disminución de la asimetría de prácticamente el 30%.

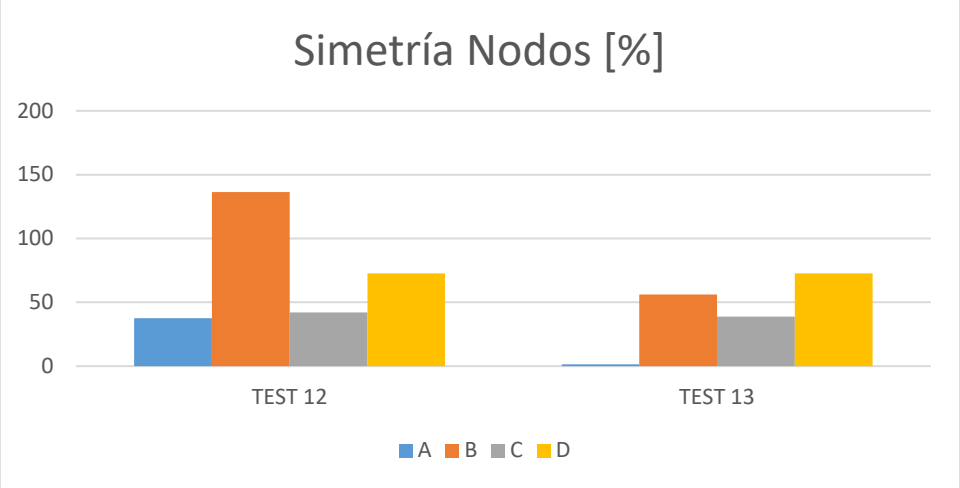


Ilustración 42. Gráfica simetría ensayo parámetros de ayuda.

Ensayo	Área total celdas [um^2]	Área total chip [um^2]	N.º Filas	Espaciado filas[um]	Ratio aspecto	Fillers añadidos	Variación media [%]
12	412.474	3781.008	3	0	0.554	3x1, 7x2, 1x4	72.2
13	412.474	3781.008	3	0	0.554	3x1, 7x2, 1x4	42.31

Tabla 10. Resumen modificación SI DRIVEN.

6.4.2 Esfuerzo

Para la comprobación del efecto de la modificación del esfuerzo se ha realizado un ensayo 14 con esfuerzo 0 (mínimo) y un ensayo 15 con esfuerzo 10 (máximo). El esfuerzo medio, sería con el nivel al 5, que es con lo que cuenta el diseño con los parámetros por defecto, por lo que no se ha vuelto a repetir.

Para el ensayo 14 con el parámetro del esfuerzo al mínimo se han obtenido exactamente los mismos resultados que poniendo este esfuerzo al nivel medio (ensayo 0) o al máximo (ensayo 15), por lo que este parámetro ya se puede decir a ciencia cierta que para este diseño no es influyente.

6.5 Posicionado manual de celdas de salida y pines

Como última opción de variación, se ha querido influir en el posicionado de pines y celdas de salida de forma manual. Esta opción es posiblemente una de las más influyentes, pero es necesario tener en cuenta que también es la que nos elimina la automatización del proceso, por lo que hay que ser conscientes de sus contras.

De este modo, se ha tratado de dejar todos los parámetros por defecto, modificando únicamente el posicionado de los pines de entrada/salida, acomodándolos lo más cercano posible entre las señales y sus complementarias, con la intención de reducir al mínimo el producto RC.

Para este ensayo 16, los resultados de variación del producto RC entre nodos obtenidos son de 48.96% para el nodo A, 167.65% para el nodo B, 7.4% para el nodo C y 0% para el nodo D, lo que hace una variación media del 56%.

A simple vista puede verse como la variación media es mayor que en algunos casos. Sin embargo, el valor medio es próximo al del ensayo 0 y no existen picos excesivamente elevados que desvirtúen este valor medio, como sucede en alguno de los ensayos anteriores llegando a nodos de salida con una asimetría de más del 600%.

Tras esto, se ha realizado un ensayo 17 en el que además de mover los pines de entrada/salida, se han movido las celdas. Así, la disposición que se ha seguido ha sido la de colocar las celdas de las señales de salida cercanas entre las de los mismos nodos y sus complementarios y en la parte exterior del diseño, y cerca de estos los pines de salida, de modo que se facilita el routing de estas señales y sus complementadas. Tras realizar un estudio preliminar y alguna prueba de ejemplo sencilla, se ha llegado al diseño que se muestra en la Ilustración 44. En esta ilustración, fácilmente se observa como las señales de salida tienen unos caminos muy similares, y en el mismo metal (otra de las condiciones impuestas). Con este diseño los resultados de variación del producto RC entre nodos obtenidos son de 52.63% para el nodo A, 44% para el nodo B, 9.09% para el nodo C y 0% para el nodo D, lo que hace una variación media del 24.16%, es decir, la variación media más baja de todos los ensayos realizados.

Se ha realizado un resumen gráfico de este bloque en la Ilustración 43, así como un resumen de los parámetros más importantes en la Tabla 11.

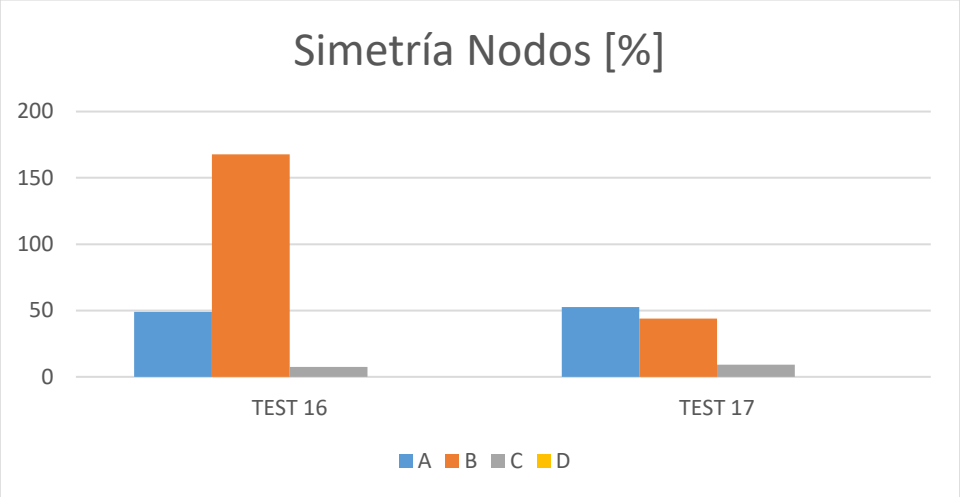


Ilustración 43. Posicionado manual.

Ensayo	Área total celdas [um^2]	Área total chip [um^2]	N.º Filas	Espaciado filas[um]	Ratio aspecto	Fillers añadidos	Variación media [%]
16	412.474	3781.008	3	0	0. 554	7x1, 5x2, 1x4	56
17	412.474	3781.008	3	0	0.554	3x1, 5x2, 2x4	24.16

Tabla 11. Resumen modificación pines y celdas manualmente.

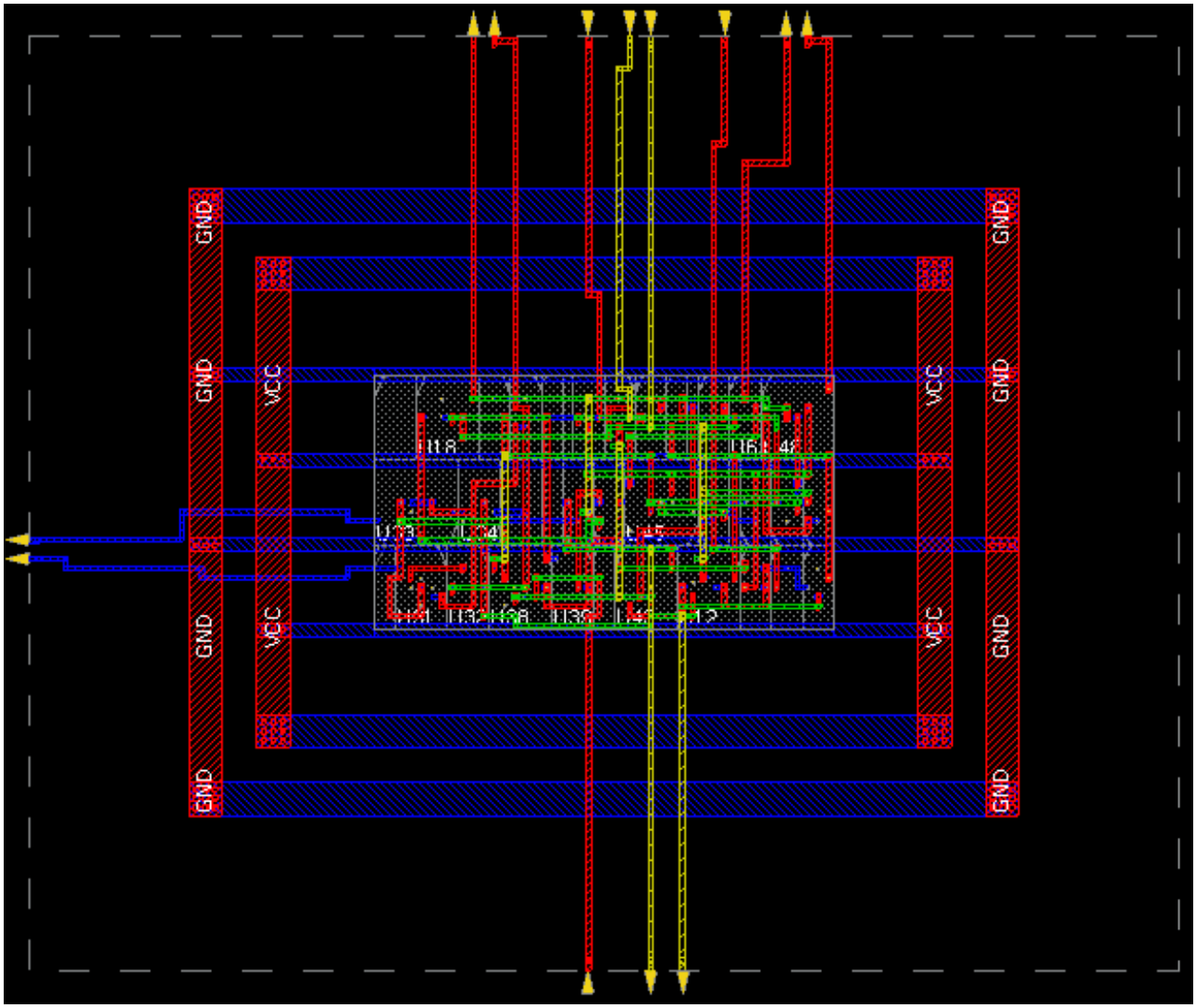


Ilustración 44. Diseño con celdas y pines posicionados manualmente.

6.6 Análisis de resultados

Tras la realización de diversos ensayos y variaciones de los mencionados parámetros, se han obtenido varios e interesantes resultados que son importantes de analizar para ver cuales parámetros son los más influyentes, en qué manera, los posibles motivos, y si son aplicables para todos los diseños o si dependen más bien del diseño con el que estemos trabajando. Por ello se ha realizado una comparativa de los resultados y un posterior análisis, resumido todo en la Tabla 12 para obtener una visión general sencilla.

En dependencia del parámetro que se ha modificado, se ha organizado la Tabla 12 por colores. Se ha incluido la variación media del producto RC y qué diseños serían comparables, de modo que, por ejemplo, el diseño con los parámetros por defecto, es comprable a gran parte de los ensayos. Los ensayos que tienen el mismo parámetro modificado, pero a diferente valor se comparan entre ellos, e incluso los de diferentes parámetros, pero con un mismo valor en común, como puede ser la separación entre filas, también son comparables (ejemplo del (6) y (7)).

Ensayo	Parámetro modificado	Variación media producto RC (%)	Ensayos comparables*
0	Ensayo por defecto.	42.31	(0)
1	Número de filas: 4	109.66	(0), (1)
2	Número de filas: 3	42.31	(0), (1)
3	Número de filas: 2	46.56	(0), (1)
4	Número de filas: 1	177.77	(0), (1)
5	Espaciado de filas: 1.12	186.68	(2)
6	Espaciado de filas: 2.24	155.97	(2), (6)
7	Espaciado de filas: 3.36	105.65	(2)
8	Espaciado de filas: 4.48	47.65	(2), (7)
9	Espaciado de filas: 5.6	35.08	(2)
10	Double-back rows top-to-bottom y separación: 2.24	102.07	(0), (3), (6)
11	Double-back rows top-to-bottom y separación: 4.48	78.57	(0), (3), (7)
12	SI Driven: no	72.2	(0), (4)
13	SI Driven: sí	42.31	(0), (4)
14	Esfuerzo: 0	42.31	(0), (5)
15	Esfuerzo: 10	42.31	(0), (5)
16	Posicionado manual: Pines	56	(0), (8)
17	Posicionado manual: Pines y celdas	24.16	(0), (8)

Tabla 12. Comparativa de los resultados para diferentes parámetros.

* Los ensayos que tienen el mismo número son comparables entre ellos, así todos los que tengan un (0) u otro número se podrán comparar entre sí (que tengan el número (0) en ensayo comparable no tiene por qué hacer referencia a que se comparen con el ensayo 0, aunque puede coincidir, como es el caso. Por ejemplo, los ensayos comparables (8), no significa que sean comparable con el ensayo 8, si no que los dos que tienen el (8), son comparables entre ellos, como ocurre con los ensayos 16 y 17.

Si se analizan los datos obtenidos respecto al número de filas, se observa como de entre las cuatro opciones existentes, la mejor parada para este diseño es la correspondiente a 3 filas, curiosamente coincidiendo con la elección automática del programa cuando se determinan los parámetros por defecto. Por lo tanto, la primera conclusión que se puede sacar es que, para este diseño en concreto, la mejor simetría para las señales de salida se obtiene cuando el floorplan está compuesto por 3

filas, seguido por el diseño con 2 filas, y obteniendo resultados poco simétricos respecto de lo que podemos alcanzar con el diseño con 1 y 4 filas.

Respecto a la influencia de la separación entre filas, estas pruebas solo pueden compararse entre ellas ya que cuando se selecciona una separación entre filas, el posicionado de los pines varía y, por tanto, no se pueden comparar con el resto de ensayos. Así, se observa cómo cuanto mayor es la separación entre filas, mejores son los resultados que se obtienen. Esto tiene sentido y coincide con lo que se esperaba respecto a este parámetro ya que, a mayor separación entre filas, menos densidad de pistas hay en el diseño y, por tanto, más facilidad de routing y simetría puede haber. Sin embargo, si además de la simetría se miran otros parámetros como el tamaño del diseño, obviamente esto se verá empeorado, por lo que hay que buscar la relación óptima. Por ejemplo, para el caso en el que se ha ido variando el espaciado, el área del chip ha aumentado, de modo que por un lado se ha mejorado respecto simetría de señales, pero por otro, se ha aumentado el área del chip. Esto se puede ver representado en la Ilustración 45.

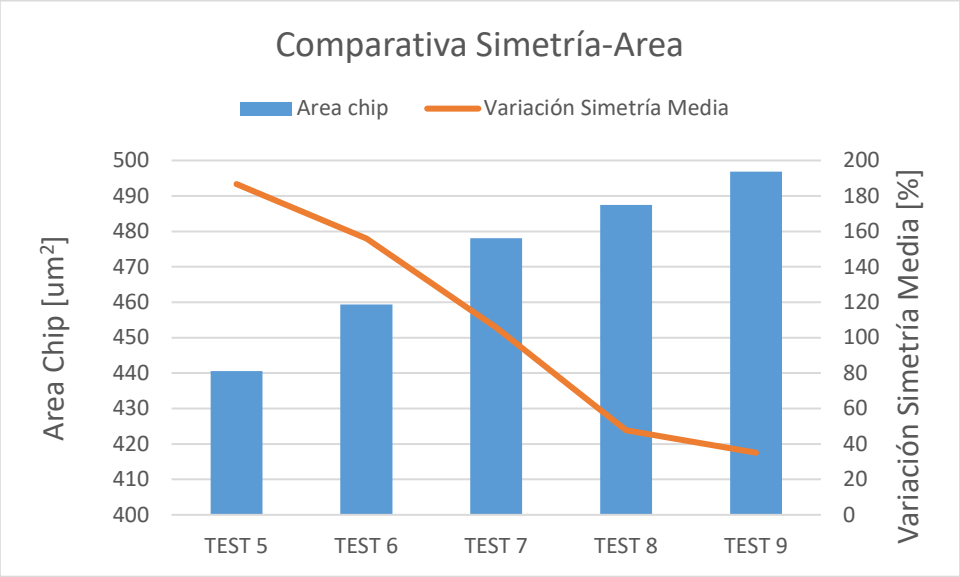


Ilustración 45. Comparativa Simetría-Área

Otras de las opciones que se ha barajado ha sido la variación de la posición relativa de la masa y la alimentación (parámetro *Double-back rows*). En este caso hay una pequeña discrepancia y no sigue un camino ya que, en el caso en el que se modifica este parámetro de como viene por defecto (en referencia al posicionado de la alimentación y la masa) y la separación es 2.24 um, se obtienen unos resultados mejores que en el parámetro por defecto. Sin embargo, para cuando esto se realiza con una separación de 4.48 um, se obtienen peores resultados que cuando se deja este parámetro por defecto. Por lo tanto, podemos establecer que es un parámetro influyente, pero no es generalizable, ya que dependerá del diseño en concreto que se esté trabajando para saber cuál de los dos estilos es más beneficioso.

Centrándonos en los parámetros de ayuda se ha diferenciado entre SI driven y el esfuerzo en el routing. Para el primero de ellos vemos como el hecho de activarlo, reduce la variación media del producto RC, por lo que las salidas son más simétricas (prácticamente el doble). De este parámetro se puede sacar como conclusión que es bastante efectivo a la hora de realizar la mejora en el diseño, incluso para un diseño con una cantidad de puertas lógicas no muy elevado. Por el contrario, el hecho de modificar el esfuerzo de routing no afecta a los resultados finales. Esto puede darse debido

a que las restricciones temporales de este diseño son mínimas, por lo que con el mínimo esfuerzo ya es posible cumplir lo necesario.

Por último, podemos ver como el parámetro que más influencia los resultados finales del diseño es el de posicionar manualmente los pines y las celdas. El hecho de mover únicamente los pines es muy dependiente del diseño y de cómo se haya realizado el placement de las celdas, ya que por ejemplo para este diseño, al posicionar los pines de salida y sus complementarios cerca unos de otros, se conseguía que la capacitancia (C), fuera prácticamente igual en ambas líneas. Sin embargo, el valor de resistencia (R), se comportaba muy diferente, debido a las longitudes de las líneas y las vías necesarias para alcanzar la posición definida. No obstante, cuando se realiza el posicionamiento manualmente tanto de los pines de salida como de las celdas, la situación cambia. En este diseño, y previsiblemente aplicable a todos, una buena colocación de celdas y pines ayuda en gran medida a la simetría. El hecho de seleccionar cuáles son las celdas de las que provienen las salidas, posicionarlas cercas entre ellas y en la parte más exterior del diseño y acompañar esto con el correcto posicionado de pines y selección del metal, ha sido el ensayo con el que mejores beneficios se han obtenido, llegando a alcanzar una reducción de la variación del producto RC con parámetros estándar del 42.91%, casi a la mitad. (De 42.31% a 24.16%).

Las modificaciones de este último ensayo 17 son las que mejores resultados presentan, pero también son las más costosas, ya que elimina la opción de automatizar el proceso. Además, cabe resaltar que llega un punto en el que reducir más la variación es muy complejo y no resulta rentable, ya que puede darse la opción de dedicarle muchas horas (en función de la complejidad del diseño), para una variación de menos de un punto porcentual. Esto se debe a la naturaleza de las señales y la interconexión de las salidas del circuito con celdas intermedias, a modo de realimentación, como ocurre con las señales A y B en esta S-Box, que, al realimentar a celdas intermedias, las líneas son relativamente largas, y el valor de la resistencia se ve incrementado, aumentando el producto RC y con ello la variación media.

7 METODOLOGÍAS Y HERRAMIENTAS EMPLEADAS

La metodología aplicada para la realización del trabajo ha sido determinante para la buena organización y finalización del trabajo. Esta se ha aplicado de la manera que se ha instado a tomar desde el comienzo del master y más centrado en la parte de CAD, como se aprende durante la asignatura de Metodologías y Herramientas CAD, principal fuente de aprendizaje y conocimientos en lo que metodología y herramientas se refiere para este trabajo.

El comienzo del trabajo se ha basado en el estudio del marco teórico. Para ello se ha realizado una investigación bibliográfica con el fin de disponer de la mayor cantidad de información relevante sobre el tema para poder abordar con conocimiento y experiencia de otros autores el contenido en el que se ha trabajado.

Una vez se ha recabado información y no solo se ha leído, sino que se ha comprendido, ha sido el momento de pasar a la acción y lo que se puede considerar como parte práctica del trabajo, comenzando por la realización del código tanto de la lógica de la S-box 4 y la etapa de precarga y evaluación, como la confección de un testbench empleado para poner a prueba la calidad del código, utilizando para ello un lenguaje HDL, en este caso Verilog, y una de las herramientas utilizadas durante este máster para ello, Modelsim SE 6.6d. La tecnología objeto del trabajo es UMC 180nm, disponible en las librerías del Máster.

Tras realizar la lógica, se ha utilizado el programa Design Vision G-2012.06-SP4 como herramienta para la conversión del código a formato de esquemático y para realizar la síntesis lógica. Con ello, además de la síntesis se ha obtenido información como la frecuencia máxima ideal del circuito teniendo en cuenta el retraso de las puertas, el área de las celdas, cantidad y tipo de estas y el fichero de retrasos (SDF).

A continuación, se ha vuelto a realizar en Modelsim la comprobación del circuito sintetizado con el testbench, demostrando que se incluyen los retrasos causados por las puertas y aun así el funcionamiento del circuito es el correcto, permitiendo con ello avanzar en el proceso y centrarse en la realización de la parte física.

En lo que respecta al layout, se ha comenzado con la realización de un diseño con parámetros considerados estándar según la referencia [22], y cuyos parámetros de placement y layout se han tomado como base de partida. Como método de realizar diferentes ensayos se ha escogido la opción de variar los parámetros de Place & Route de forma aislada, con el motivo de ver la influencia de cada parámetro por separado, consiguiendo una realización de 18 ensayos en total y obteniendo los datos mediante la herramienta de diseño INNOVUS 15.20.000.

Posteriormente se ha realizado el análisis y procesado de los datos de los diseños, enfocando el esfuerzo mayormente en lo que a resistencia y capacidad del nodo de salida respecta, para obtener así el producto RC (medidor de simetría considerado), la variación relativa del nodo complementado respecto del original, y la variación media del total de los nodos de salida, empleando para todo esto un documento Excel con una macro creada y personalizada para este fin.

Por último, se ha realizado una comparativa de los datos obtenidos y se han sacado las conclusiones de estos, permitiendo conocer la influencia de los diferentes parámetros aplicados a este diseño en lo que a la simetría de los nodos de salida respecta.

Este proceso ha sido representado en el diagrama de bloques presente en la Ilustración 46 para mostrar de forma gráfica la metodología y las herramientas de trabajo con las que se han trabajado.

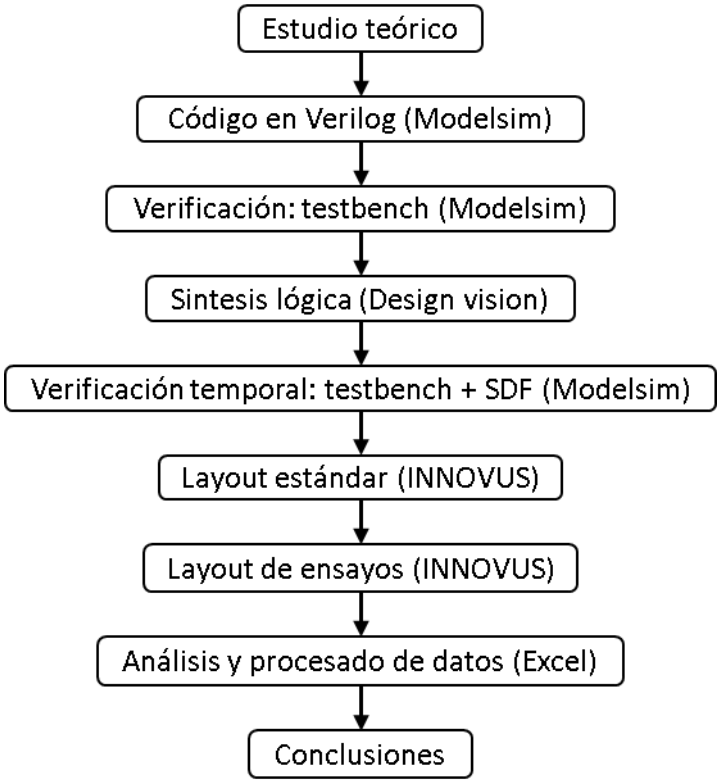


Ilustración 46. Proceso de trabajo.

8 CONCLUSIONES

El principal objetivo de este trabajo final de máster ha sido el de analizar los parámetros más influyentes en el Place & Route de un dispositivo criptográfico para determinar de qué manera se puede maximizar la simetría para uso de routing diferencial, a partir de las modificaciones del proceso previo de Place & Route.

Para ello, se ha realizado un estudio y análisis teórico de las diversas lógicas de doble rail existentes aplicables a ASICs para hacer frente a los DPAs y posteriormente se ha continuado con la investigación sobre las diferentes técnicas de routing diferencial existentes para lógicas de doble rail.

Tras haber finalizado lo que se considera el marco teórico del trabajo, se ha acometido el proceso de diseño de un ejemplo criptográfico (S-box 4 de Piccolo) aplicando una de las lógicas DPL estudiadas (WDDL), para ver, de cara a la aplicación de la técnica de layout escogida, como es el proceso previo de Place & Route y analizar la asimetría presente en sus salidas.

En este caso, se ha escogido la técnica de *Fat Wire* y se ha utilizado como unidad elemental la S-Box por las consideraciones de compromiso en la elección de granularidad que ya se han explicado en el trabajo.

Una vez se ha completado el diseño del ejemplo criptográfico, se ha llevado a cabo un estudio para saber cuáles de los parámetros previos al Place & Route son los más influyentes respecto a la simetría en las señales de salida. De este modo, se ha podido obtener información para realizar un diseño automático o semiautomático, y conocer directamente la influencia de ciertos parámetros, concluyendo que hay parámetros que afectan en según qué medida dependiendo del diseño (Número de filas en el floorplan o Variación de posicionado relativo de alimentación y masa), otros afectan mejorando la simetría a cambio de empeorar otros parámetros (Variación del espaciado de filas: aumenta el área del diseño) o incluso los que está claro que mejoran notablemente el resultado final, como Activación de parámetros de ayuda o Posicionado manual de celdas de salida y pines, aunque este último supone eliminar la automatización del proceso.

La realización del trabajo ha implicado la utilización de la mayoría de conocimientos adquiridos en la asignatura de Metodologías de Diseño y Herramientas de CAD, siguiendo todo el flujo de diseño y profundizando en muchos aspectos, así como conocimientos de otras asignaturas o de experiencias previas ya sean laborales o estudiantiles.

A título personal, me ha resultado muy interesante la realización de este trabajo, ya que además de aplicar conocimientos aprendidos durante el máster, he aprendido o profundizado otros y, sobre todo, haciendo lo que me gusta, que es el diseño hardware. Con esto me ha quedado más claro cómo pueden afectar los parámetros al realizar el diseño, y lo influyente que puede ser el hecho de desautomatizar el proceso.

Considerando el trabajo realizado, creo que se han alcanzado con creces los objetivos propuestos, a pesar de haber dejado algunas líneas de trabajo abiertas, como podría haber sido el aplicar el mismo proceso en otro ejemplo criptográfico para comparar resultados, o incluso realizar el proceso con otra lógica o técnica de layout.

9 REFERENCIAS

[1] K. Tiri et al. (2005). Prototype IC with WDDL and Differential Routing – DPA Resistance Assessment. En CHES 2005, LNCS 3659 (354-365). Springer US.

[2] S. Mangard, E. Oswald, T. Popp. (2007). Power analysis attacks: revealing the secrets of smart cards. Springer US.

[3] J. Danger, S. Guilley, S. Bhasin, M. Nassar, L. Sauvage. (2009). Overview of Dual Rail with Precharge Logic Styles to Thwart Implementation-Level Attacks on Hardware Cryptoprocessors. En Signals, Circuits and Systems (SCS), (1-8).

[4] K. Tiri, I. Verbauwhede. (2004). A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. En IEEE Computer Society 2004, (246-251). Paris, Francia.

[5] K. Tiri, M. Akmal, I. Verbauwhede. (2002). A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. En ESSCIRC, 2002, (403-406).

[6] Z. Chen, Y. Zhou. (2006). Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage. En CHES 2006, LNCS 4249 (242-254). Springer US.

[7] T. Popp, S. Mangard. (2005). Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. En CHES 2005, LNCS 3659 (172-186). Springer US.

[8] T. Popp, M. Kirschbaum, T. Zeerer, S. Mangard. (2007). Evaluation of the Masked Logic Style MDPL on a Prototype Chip. En CHES 2007, LNCS 4727 (81-94). Springer US.

[9] A. Wild, A. Moradi, T. Güneysu. (2015). Evaluating the Duplication of Dual-Rail Precharge Logics on FPGAs. En COSADE 2015, (81-94). Springer US.

[10] K. Tiri, I. Verbauwhede. (2004). A Dynamic and Differential CMOS Logic Style to Resist Power and Timing Attacks on Security IC's. IACR Cryptology ePrint Archive.

[11] B. Preneel, T. Takagi. (2011). Cryptographic Hardware and Embedded Systems -- CHES 2011. 13th International Workshop, Nara, Japan, September 28 -- October 1, 2011, Proceedings. Springer US.

[12] S. Guilley, L. Sauvage, P. Hoogvorst, R. Pacalet, G. M. Bertoni, S. Chaudhuri. (2008). Security Evaluation of WDDL and SecLib Countermeasures against Power Attacks. En IEEE Transactions on Computers, vol. 57, no. 11 (1482-1497).

[13] A. Moradi, M. Kirschbaum, T. Eisenbarth, C. Paar. (2012). Masked Dual-Rail Precharge Logic Encounters State-of-the-Art Power Analysis Methods. En IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 20, no. 9 (1578-1589).

[14] R. Soares, N. Calazans, V. Lomné, P. Maurine, L. Torres, M. Robert. (2008). Evaluating the Robustness of Secure Triple Track Logic through Prototyping. En Proceedings of the 21st Annual Symposium on Integrated Circuits and System Design (193-198).

[15] K. Tiri, I. Verbauwhede. (2004). Place and Route for Secure Standard Cell Design. En Smart Card Research and Advanced Applications VI. IFIP International Federation for Information Processing, vol 153 (143-158). Springer US.

[16] K. Tiri and I. Verbauwhede. (2006). A digital design flow for secure integrated circuits. En IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 7 (1197-1208).

[17] S. Guilley, P. Hoogvorst, Y. Mathieu, R. Pacale. (2005). The "Backend Duplication" Method. En CHES 2005, LNCS 3659 (383-397). Springer US.

[18] K. Baddam, M. Zwolinski. (2008). Divided backend duplication methodology for balanced dual rail routing. En CHES 2008, LNCS 5154 (396–410). Springer US.

[19] P. Yu, P. Schaumont. (2007). Secure FPGA circuits using controlled placement and routing. En CODES+ISSS 2007: Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis (45–50). ACM, Nueva York.

[20] P. Yu. (2007). Implementation of DPA-Resistant Circuit for FPGA. (Trabajo de master). Facultad del instituto politécnico de Virginia y Universidad del estado. Blacksburg, Virginia.

[21] Y. Li, K. Ohta, K. Sakiyama. (2011). Revisit fault sensitivity analysis on WDDL-AES. En IEEE International Symposium on Hardware-Oriented Security and Trust (148-153). San Diego, California.

[22] C.J. Jiménez Fernández. (2018). Unit 6: Innovus tutorial. Instituto de Microelectrónica de Sevilla / Universidad de Sevilla. Sevilla, España.

10 AGRADECIMIENTOS

Este trabajo de final de máster es un esfuerzo en el cual han colaborado de forma directa o indirecta distintas personas, ya haya sido ayudando, corrigiendo, aportando material, prestando consejo y/u opinión, así como dando ánimo en momentos difíciles y celebrando en los momentos felices.

En primer lugar, a mis tutores de trabajo de fin de master Don Antonio José Acosta Jiménez y Doña Erica Tena Sánchez, agradecerle el haber dedicado parte de su tiempo para tutorizar este trabajo, así como ayudarme en lo necesario, haciendo lo posible por la correcta finalización del trabajo.

A mi familia, amigos y compañeros, darles las gracias por su apoyo, ánimo y tiempo compartido conmigo. Por saber llevarme en momentos de crisis y ansiedad y estar por y para mí cuando así lo he necesitado.

11 ANEXO 1

Código verilog S-Box 4 Piccolo.

```
module SBOX4_PICOLO(a, b, c, d, A, B, C, D, AN, BN, CN, DN, CLK);
input a, b, c, d, CLK;
output A, B, C, D, AN, BN, CN, DN;
reg  ap, bp, cp, dp, an, bn, cn, dn;
wire q1p, q2p, q3p, q4p, q1n, q2n, q3n, q4n;

 always @(CLK or a or b or c or d)
 begin
  //Fase de precarga
  ap=~(CLK | (~a));
  an=~(CLK | a);

  bp=~(CLK | (~b));
  bn=~(CLK | b);

  cp=~(CLK | (~c));
  cn=~(CLK | c);

  dp=~(CLK | (~d));
  dn=~(CLK | d);
 end

 //Fase de evaluacion
 assign q1p =(an & bn);
 assign q1n =(ap | bp);

 assign q2p =(bn & cn);
 assign q2n =(bp | cp);

 assign q3p =(AN & cn);
 assign q3n =(A  | cp);

 assign q4p =(BN & AN);
 assign q4n =(B  | A);

 assign A   = ((dp & q1n) | (dn & q1p));
 assign AN  = ((dn | q1p) & (dp | q1n));

 assign B   = ((ap & q2n) | (an & q2p));
 assign BN  = ((an | q2p) & (ap | q2n));

 assign C   = ((bn | q3p) & (bp | q3n));
 assign CN  = ((bp & q3n) | (bn & q3p));

 assign D   = ((cp & q4n) | (cn & q4p));
 assign DN  = ((cn | q4p) & (cp | q4n));

endmodule
```

12 ANEXO 2

Código verilog Testbench S-Box 4 Piccolo y simulación post síntesis.

```
`timescale 1ns / 1ns

module SBOX4_PICOLO_TB();

reg CLK, a, b, c, d;
SBOX4_PICOLO DUT(.CLK(CLK), .a(a), .b(b), .c(c), .d(d), .A(A), .B(B), .C(C), .D(D), .AN(AN), .BN(BN), .CN(CN), .DN(DN));

initial
begin
CLK = 0;
forever #10 CLK=~CLK;
end

initial
begin                  // Inicializacion de parametros
a=0; b=0; c=0;d=0;

#7 d=1;
#10 c=1;d=0;
#20 d=1;

#10 b=1;c=0;d=0;
#10 d=1;
#10 c=1;d=0;
#10 d=1;

#10 a=1;b=0;c=0;d=0;
#10 d=1;
#10 c=1;d=0;
#10 d=1;

#10 b=1;c=0;d=0;
#10 d=1;
#10 c=1;d=0;
#10 d=1;

end
endmodule
```

13 ANEXO 3

Contenido del fichero de introducción manual del posicionado de pines de entrada y salida (*file.io*)

```
(globals
   version = 3
   io_order = default
)
(iopin
   (top
         (pin name="A0"    offset=26.3500 layer=2 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="AN"    offset=27.5900 layer=2 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="b"     offset=33.1700 layer=2 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="c"     offset=35.6500 layer=4 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="CLK"   offset=36.8900 layer=4 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="a"     offset=41.2300 layer=2 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="CN"    offset=44.9500 layer=2 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="C0"    offset=46.1900 layer=2 width=0.2800 depth=0.7000 place_status=placed  )
   )
   (left
         (pin name="D0"    offset=24.3600 layer=1 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="DN"    offset=25.4800 layer=1 width=0.2800 depth=0.7000 place_status=placed  )
   )
   (bottom
         (pin name="d"     offset=33.1700 layer=2 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="B0"    offset=36.8900 layer=4 width=0.2800 depth=0.7000 place_status=placed  )
         (pin name="BN"    offset=38.7500 layer=4 width=0.2800 depth=0.7000 place_status=placed  )
   )
)
```

14 ANEXO 4

Contenido del fichero Excel empleado para el procesado y análisis de datos obtenidos de INNOVUS para la obtención del producto RC, variación relativa al nodo original y variación relativa media del ensayo.

Ensayo 0								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	72,72727273
set_load	0,008	"A0"	set_resistance	0,063	"A0"	RC->	0,000504	1,388888889
set_load	0,007	"AN"	set_resistance	0,071	"AN"	RC->	0,000497	0
set_load	0,005	"B0"	set_resistance	0,053	"B0"	RC->	0,000265	56,22641509
set_load	0,006	"BN"	set_resistance	0,069	"BN"	RC->	0,000414	0
set_load	0,003	"C0"	set_resistance	0,018	"C0"	RC->	0,000054	38,88888889
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"DN"	set_resistance	0,019	"DN"	RC->	0,000057	0
Variación relativa media [%] ->								**42,3078664**
Ensayo 1								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	215,1515152
set_load	0,004	"DN"	set_resistance	0,026	"DN"	RC->	0,000104	0
set_load	0,003	"CN"	set_resistance	0,018	"CN"	RC->	0,000054	0
set_load	0,003	"C0"	set_resistance	0,018	"C0"	RC->	0,000054	0
set_load	0,009	"BN"	set_resistance	0,064	"BN"	RC->	0,000576	0
set_load	0,005	"B0"	set_resistance	0,04	"B0"	RC->	0,0002	188
set_load	0,008	"AN"	set_resistance	0,079	"AN"	RC->	0,000632	0
set_load	0,01	"A0"	set_resistance	0,098	"A0"	RC->	0,00098	35,51020408
Variación relativa media [%] ->								**109,6654298**
Ensayo 2								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	72,72727273
set_load	0,003	"DN"	set_resistance	0,019	"DN"	RC->	0,000057	0
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"C0"	set_resistance	0,018	"C0"	RC->	0,000054	38,88888889
set_load	0,006	"BN"	set_resistance	0,069	"BN"	RC->	0,000414	0
set_load	0,005	"B0"	set_resistance	0,053	"B0"	RC->	0,000265	56,22641509
set_load	0,007	"AN"	set_resistance	0,071	"AN"	RC->	0,000497	0
set_load	0,008	"A0"	set_resistance	0,063	"A0"	RC->	0,000504	1,388888889
Variación relativa media [%] ->								**42,3078664**

Ensayo 3								
Variable	Carga	Nodo	Variable	Carga	Nodo		Producto RC	Variación relativa [%]
set_load	0,003	"DN"	set_resistance	0,012	"DN"	RC->	0,000036	0
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	9,090909091
set_load	0,003	"CN"	set_resistance	0,012	"CN"	RC->	0,000036	0
set_load	0,003	"C0"	set_resistance	0,011	"C0"	RC->	0,000033	9,090909091
set_load	0,008	"BN"	set_resistance	0,065	"BN"	RC->	0,00052	0
set_load	0,006	"B0"	set_resistance	0,043	"B0"	RC->	0,000258	101,5503876
set_load	0,005	"AN"	set_resistance	0,041	"AN"	RC->	0,000205	0
set_load	0,009	"A0"	set_resistance	0,068	"A0"	RC->	0,000612	66,50326797
Variación relativa media [%] ->								**46,55886844**

Ensayo 4								
Variable	Carga	Nodo	Variable	Carga	Nodo		Producto RC	Variación relativa [%]
set_load	0,003	"DN"	set_resistance	0,011	"DN"	RC->	0,000033	0
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	0
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"C0"	set_resistance	0,011	"C0"	RC->	0,000033	0
set_load	0,008	"BN"	set_resistance	0,052	"BN"	RC->	0,000416	0
set_load	0,003	"B0"	set_resistance	0,019	"B0"	RC->	0,000057	629,8245614
set_load	0,005	"AN"	set_resistance	0,054	"AN"	RC->	0,00027	0
set_load	0,015	"A0"	set_resistance	0,096	"A0"	RC->	0,00144	81,25
Variación relativa media [%] ->								**177,7686404**

Ensayo 5								
Variable	Carga	Nodo	Variable	Carga	Nodo		Producto RC	Variación relativa [%]
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,004	"DN"	set_resistance	0,026	"DN"	RC->	0,000104	0
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	215,1515152
set_load	0,004	"C0"	set_resistance	0,012	"C0"	RC->	0,000048	31,25
set_load	0,007	"BN"	set_resistance	0,061	"BN"	RC->	0,000427	0
set_load	0,004	"B0"	set_resistance	0,019	"B0"	RC->	0,000076	461,8421053
set_load	0,009	"AN"	set_resistance	0,064	"AN"	RC->	0,000576	0
set_load	0,009	"A0"	set_resistance	0,104	"A0"	RC->	0,000936	38,46153846
Variación relativa media [%] ->								**186,6762897**

Ensayo 6								
Variable	Carga	Nodo	Variable	Carga	Nodo		Producto RC	Variación relativa [%]
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,004	"C0"	set_resistance	0,013	"C0"	RC->	0,000052	36,53846154
set_load	0,004	"DN"	set_resistance	0,026	"DN"	RC->	0,000104	0
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	215,1515152
set_load	0,007	"BN"	set_resistance	0,062	"BN"	RC->	0,000434	0
set_load	0,005	"B0"	set_resistance	0,02	"B0"	RC->	0,0001	334
set_load	0,008	"AN"	set_resistance	0,057	"AN"	RC->	0,000456	0
set_load	0,006	"A0"	set_resistance	0,055	"A0"	RC->	0,00033	38,18181818
Variación relativa media [%] ->								**155,9679487**

Ensayo 7								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,004	"DN"	set_resistance	0,026	"DN"	RC->	0,000104	0
set_load	0,003	"D0"	set_resistance	0,025	"D0"	RC->	0,000075	38,66666667
set_load	0,004	"C0"	set_resistance	0,039	"C0"	RC->	0,000156	78,84615385
set_load	0,008	"BN"	set_resistance	0,063	"BN"	RC->	0,000504	0
set_load	0,004	"B0"	set_resistance	0,033	"B0"	RC->	0,000132	281,8181818
set_load	0,008	"AN"	set_resistance	0,07	"AN"	RC->	0,00056	0
set_load	0,01	"A0"	set_resistance	0,073	"A0"	RC->	0,00073	23,28767123
Variación relativa media [%] ->								**105,6546684**

Ensayo 8								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,011	"A0"	set_resistance	0,072	"A0"	RC->	0,000792	11,61616162
set_load	0,01	"AN"	set_resistance	0,07	"AN"	RC->	0,0007	0
set_load	0,007	"B0"	set_resistance	0,062	"B0"	RC->	0,000434	11,98156682
set_load	0,009	"BN"	set_resistance	0,054	"BN"	RC->	0,000486	0
set_load	0,005	"C0"	set_resistance	0,02	"C0"	RC->	0,0001	67
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"D0"	set_resistance	0,018	"D0"	RC->	0,000054	100
set_load	0,004	"DN"	set_resistance	0,027	"DN"	RC->	0,000108	0
Variación relativa media [%] ->								**47,64943211**

Ensayo 10								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,004	"C0"	set_resistance	0,013	"C0"	RC->	0,000052	36,53846154
set_load	0,007	"A0"	set_resistance	0,056	"A0"	RC->	0,000392	42,85714286
set_load	0,008	"AN"	set_resistance	0,07	"AN"	RC->	0,00056	0
set_load	0,005	"B0"	set_resistance	0,02	"B0"	RC->	0,0001	140
set_load	0,008	"BN"	set_resistance	0,03	"BN"	RC->	0,00024	0
set_load	0,003	"D0"	set_resistance	0,012	"D0"	RC->	0,000036	188,8888889
set_load	0,004	"DN"	set_resistance	0,026	"DN"	RC->	0,000104	0
Variación relativa media [%] ->								**102,0711233**

Ensayo 9								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,009	"A0"	set_resistance	0,084	"A0"	RC->	0,000756	77,77777778
set_load	0,006	"AN"	set_resistance	0,028	"AN"	RC->	0,000168	0
set_load	0,006	"B0"	set_resistance	0,041	"B0"	RC->	0,000246	43,08943089
set_load	0,008	"BN"	set_resistance	0,044	"BN"	RC->	0,000352	0
set_load	0,003	"C0"	set_resistance	0,018	"C0"	RC->	0,000054	11,11111111
set_load	0,003	"CN"	set_resistance	0,02	"CN"	RC->	0,00006	0
set_load	0,003	"D0"	set_resistance	0,012	"D0"	RC->	0,000036	8,333333333
set_load	0,003	"DN"	set_resistance	0,011	"DN"	RC->	0,000033	0
Variación relativa media [%] ->								**35,07791328**

Ensayo 11								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,01	"A0"	set_resistance	0,08	"A0"	RC->	0,0008	16,25
set_load	0,01	"AN"	set_resistance	0,067	"AN"	RC->	0,00067	0
set_load	0,007	"B0"	set_resistance	0,036	"B0"	RC->	0,000252	89,28571429
set_load	0,009	"BN"	set_resistance	0,053	"BN"	RC->	0,000477	0
set_load	0,004	"C0"	set_resistance	0,02	"C0"	RC->	0,00008	58,75
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"D0"	set_resistance	0,018	"D0"	RC->	0,000054	150
set_load	0,005	"DN"	set_resistance	0,027	"DN"	RC->	0,000135	
Variación relativa media [%] ->								**78,57142857**
Ensayo 12								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	72,72727273
set_load	0,003	"DN"	set_resistance	0,019	"DN"	RC->	0,000057	0
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"C0"	set_resistance	0,018	"C0"	RC->	0,000054	38,88888889
set_load	0,006	"BN"	set_resistance	0,069	"BN"	RC->	0,000414	0
set_load	0,005	"B0"	set_resistance	0,053	"B0"	RC->	0,000265	56,22641509
set_load	0,007	"AN"	set_resistance	0,071	"AN"	RC->	0,000497	0
set_load	0,008	"A0"	set_resistance	0,063	"A0"	RC->	0,000504	1,388888889
Variación relativa media [%] ->								**42,3078664**
Ensayo 13								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	72,72727273
set_load	0,003	"DN"	set_resistance	0,019	"DN"	RC->	0,000057	0
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"C0"	set_resistance	0,019	"C0"	RC->	0,000057	42,10526316
set_load	0,006	"BN"	set_resistance	0,067	"BN"	RC->	0,000402	0
set_load	0,005	"B0"	set_resistance	0,034	"B0"	RC->	0,00017	136,4705882
set_load	0,007	"AN"	set_resistance	0,077	"AN"	RC->	0,000539	0
set_load	0,007	"A0"	set_resistance	0,056	"A0"	RC->	0,000392	37,5
Variación relativa media [%] ->								**72,20078103**
Ensayo 14								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	72,72727273
set_load	0,003	"DN"	set_resistance	0,019	"DN"	RC->	0,000057	0
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"C0"	set_resistance	0,018	"C0"	RC->	0,000054	38,88888889
set_load	0,006	"BN"	set_resistance	0,069	"BN"	RC->	0,000414	0
set_load	0,005	"B0"	set_resistance	0,053	"B0"	RC->	0,000265	56,22641509
set_load	0,007	"AN"	set_resistance	0,071	"AN"	RC->	0,000497	0
set_load	0,008	"A0"	set_resistance	0,063	"A0"	RC->	0,000504	1,388888889
Variación relativa media [%] ->								**42,3078664**

Ensayo 15								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"D0"	set_resistance	0,011	"D0"	RC->	0,000033	72,72727273
set_load	0,003	"DN"	set_resistance	0,019	"DN"	RC->	0,000057	0
set_load	0,003	"CN"	set_resistance	0,011	"CN"	RC->	0,000033	0
set_load	0,003	"C0"	set_resistance	0,018	"C0"	RC->	0,000054	38,88888889
set_load	0,006	"BN"	set_resistance	0,069	"BN"	RC->	0,000414	0
set_load	0,005	"B0"	set_resistance	0,053	"B0"	RC->	0,000265	56,22641509
set_load	0,007	"AN"	set_resistance	0,071	"AN"	RC->	0,000497	0
set_load	0,008	"A0"	set_resistance	0,063	"A0"	RC->	0,000504	1,388888889
Variación relativa media [%] ->								**42,3078664**

Ensayo 16								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"DN"	set_resistance	0,012	"DN"	RC->	0,000036	0
set_load	0,003	"D0"	set_resistance	0,012	"D0"	RC->	0,000036	0
set_load	0,005	"CN"	set_resistance	0,029	"CN"	RC->	0,000145	0
set_load	0,005	"C0"	set_resistance	0,027	"C0"	RC->	0,000135	7,407407407
set_load	0,007	"BN"	set_resistance	0,065	"BN"	RC->	0,000455	0
set_load	0,005	"B0"	set_resistance	0,034	"B0"	RC->	0,00017	167,6470588
set_load	0,006	"AN"	set_resistance	0,049	"AN"	RC->	0,000294	0
set_load	0,009	"A0"	set_resistance	0,064	"A0"	RC->	0,000576	48,95833333
Variación relativa media [%] ->								**56,00319989**

Ensayo 17								
Variable	**Carga**	**Nodo**	**Variable**	**Carga**	**Nodo**		**Producto RC**	**Variación relativa [%]**
set_load	0,003	"D0"	set_resistance	0,007	"D0"	RC->	0,000021	0
set_load	0,003	"DN"	set_resistance	0,007	"DN"	RC->	0,000021	0
set_load	0,007	"A0"	set_resistance	0,076	"A0"	RC->	0,000532	52,63157895
set_load	0,006	"AN"	set_resistance	0,042	"AN"	RC->	0,000252	0
set_load	0,005	"B0"	set_resistance	0,04	"B0"	RC->	0,0002	44
set_load	0,006	"BN"	set_resistance	0,048	"BN"	RC->	0,000288	0
set_load	0,003	"C0"	set_resistance	0,011	"C0"	RC->	0,000033	9,090909091
set_load	0,003	"CN"	set_resistance	0,012	"CN"	RC->	0,000036	0
Variación relativa media [%] ->								**24,15789474**